Classical Features - Bone Object Detection

#1 Bone Texture

Fine granularity indicates the trabecular bone structure. Differences in texture between vertebrae and surrounding soft tissue.

#2 Bone Shape

Characteristic shapes of individual vertebrae (e.g., rectangular or wedge-shaped). Curvature of the spine (lordosis, kyphosis).

#3 Bone Edge

Clear delineation of vertebral edges. Identification of intervertebral disc spaces. Needs to be made clear by labelling/outlining.

#4 Bone Size

Relative size of vertebrae (e.g., cervical vs. lumbar). Proportions of vertebrae compared to each other can help identify which part of the spine we are looking at.

#5 Landmarks

Identification of key anatomical landmarks (e.g., sacrum, coccyx). Recognition of the cervical, thoracic, and lumbar regions. This would require labelling images beforehand.

Classical Features - Bone Density Calculations

#1 Grayscale Intensity

Bone density can be inferred from the pixel intensity values; denser bone appears lighter (higher intensity) on X-rays.

#2 Edge Sharpness

The sharpness of the edges of bones can indicate density; sharper edges often correlate with denser bone.

#3 Morphometric Measurements

Calculating the ratios of cortical to trabecular bone area can help estimate overall density.

#4 Bone Area/Dimensions

Necessary in density calculations to estimate mass.

#5 Edge Sharpness

Assess the clarity of bone edges; sharper edges can indicate higher density.

Computer Vision Libraries - Testing

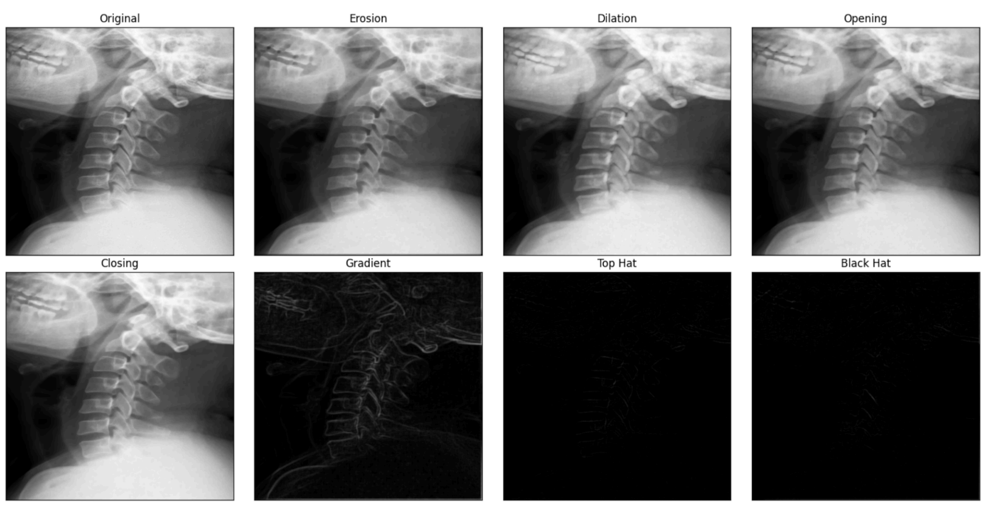
Pyradiomics

…

OpenCV

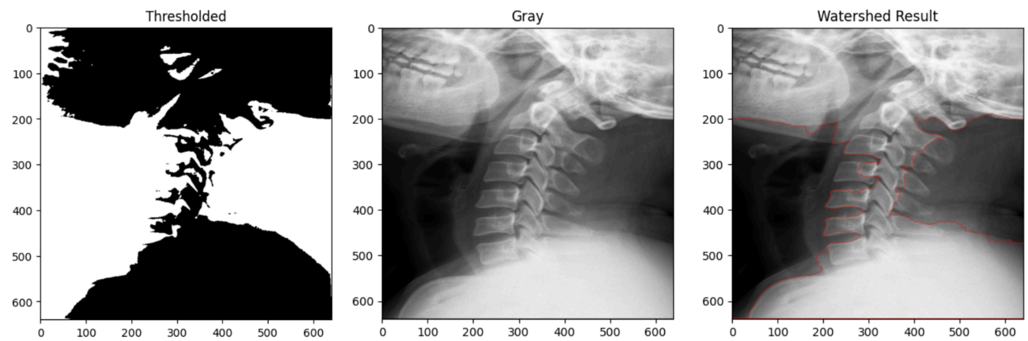
OpenCV had a few algorithms that worked for edge detection and segmentation for X-rays, but it was useful for image transformations and processing.

- ***Morphological Transformations***
  - Morphological transformations are some simple operations based on the image shape. It is normally performed on binary images. It needs two inputs, one is our original image, and the second is called the structuring element or kernel, which decides the nature of the operation. Two basic morphological operators are Erosion and Dilation. Then, its variant forms like Opening, Closing, and Gradient also come into play.
  - Gradient had the most success with edge detection and outlining, and can help with processing images to be trained. Gradient = Dilation - Erosion



- ***Watershed Algorithm - Image Segmentation***
  - The watershed algorithm views a grayscale image as a topographic surface, where high intensity represents peaks and low intensity represents valleys. When filling these valleys with colored water, barriers are built where the water from different valleys merges, leading to segmentation. However, this method can result in oversegmentation due to noise. To address this, OpenCV uses a marker-based watershed algorithm, allowing for more controlled segmentation. Users label known foreground objects with one color,

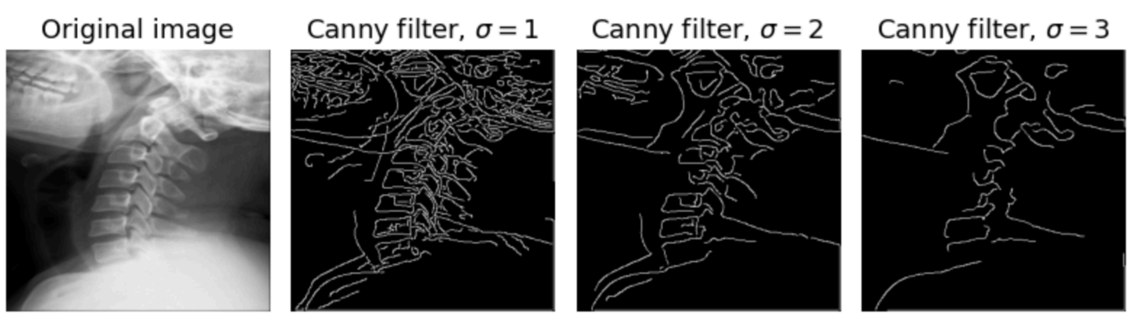
background with another, and uncertain areas with zero. After applying the watershed algorithm, the markers are updated, and object boundaries are marked with a value of -1, improving segmentation accuracy.
    - The watershed algorithm does an okay job outlining parts, but segmentation is not done correctly/well



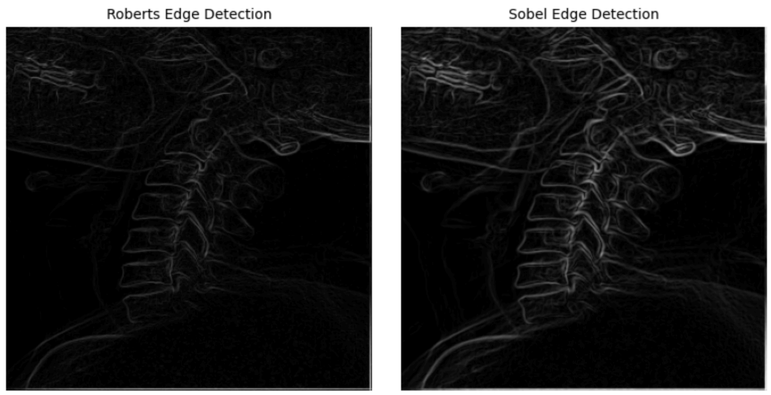
[Scikit-Image]

Scikit had a couple of algorithms that worked for edge detection and segmentation for X-rays.

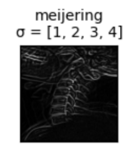
- ***[Canny Edge Detector]***
    - The Canny filter detects edges in images through several steps. It first smooths the image with a Gaussian filter to reduce noise. Then, it finds edges by calculating gradients and thins them to one-pixel lines by removing weaker pixels. Finally, it uses hysteresis thresholding to decide which edges to keep or discard based on their strength.
    - This will create an outline of the X-ray; it has a sigma parameter that can be adjusted based on how much noise should be accepted by the detector.

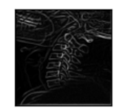


- ***[Edge Operators]***
    - They are used in image processing within edge detection algorithms. They are discrete differentiation operators that compute an approximation of the gradient of the image intensity function.
    - This also did a better job of creating an outline of the X-ray. The Sobel Edge Detection worked better than the Roberts, but both detectors show sketchy outlines instead of rigid ones.

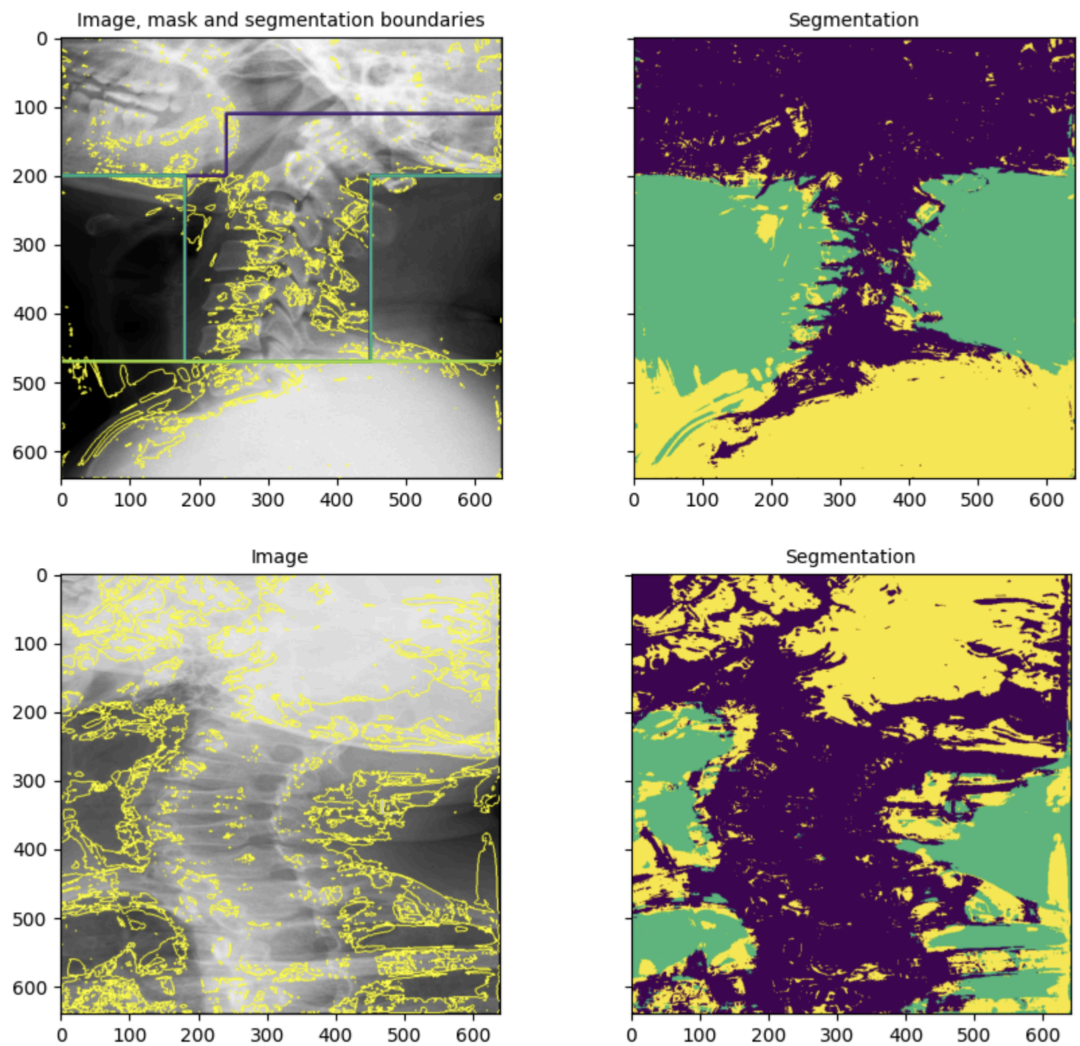


- ***Ridge Operators***
  - Ridge filters can be used to detect ridge-like structures, such as neurites, tubes, vessels, wrinkles, or rivers. Different ridge filters may be suited for detecting different structures, e.g., depending on contrast or noise level. The present class of ridge filters relies on the eigenvalues of the Hessian matrix of image intensities to detect ridge structures where the intensity changes perpendicular but not along the structure.
  - Meijering σ = [1, 2, 3, 4] had the best result.





- ***Trainable Segmentation Model (Random Forests)***
  - A pixel-based segmentation is computed here using local features based on local intensity, edges, and textures at different scales. A user-provided mask is used to identify different regions. The pixels of the mask are used to train a random-forest classifier from scikit-learn. Unlabeled pixels are then labeled from the prediction of the classifier.
  - This model did a good job at segmenting out different parts of the cervical vertebrae, splitting the skull, spine, and the torso. It requires the developer to manually create predefined masks and bounding boxes; therefore, the model only works on X-ray images that are consistent in terms of body part and positioning. It would work on a cleaner dataset; it has some issues with the current dataset.


Image, mask and segmentation boundaries
Segmentation
Image
Segmentation
0
100
200
300
400
500
600