**Image Processing of Damaged X-Ray Images**

**James Duxbury (bdzz75)**

## **Overview**

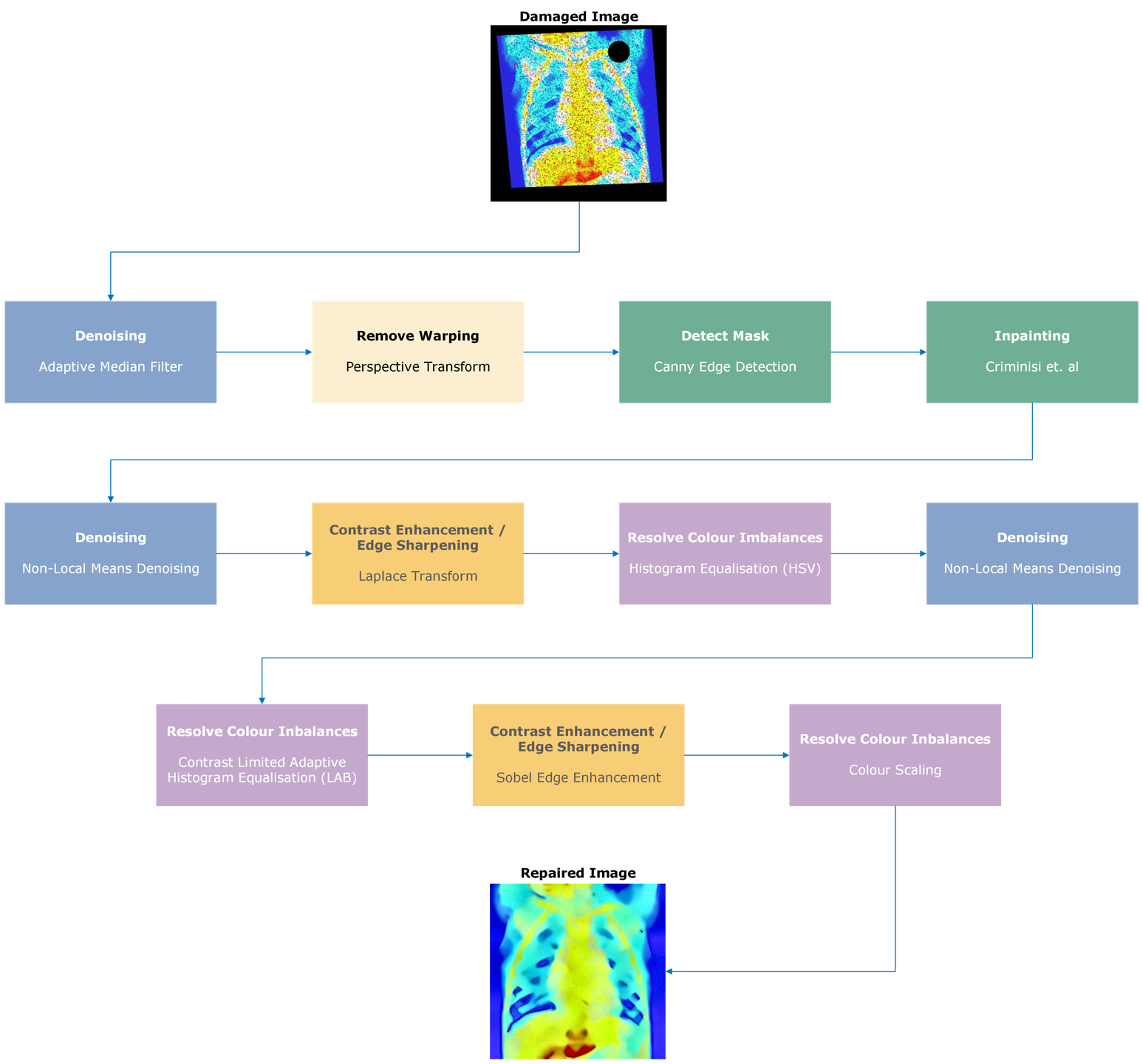
This report examines the application of image processing techniques to a dataset comprising 100 coloured X-ray images of lungs, afflicted with significant damage. The objective was to improve both the images visually, as well as the performance of a classifier, which achieved an accuracy of 0.55 before processing. The analysis investigates the impact of preprocessing steps on the classifier's accuracy, including both overall performance and individual metrics.

Fig. 1 - Overview of Image Processing Process

## 2.0 Image Processing

### 2.1 Adaptive Median Filter (AMF)

Significant amounts of Gaussian and salt-and-pepper noise were present in all images, presenting a challenge in balancing noise removal with preserving image details. While salt-and-pepper noise is typically addressed with a median filter, OpenCV's MedianBlur() function was found to induce excessive blurring and loss of details during testing.

To overcome this, a custom Adaptive Median Filter (AMF) function was implemented. This function, applied to each colour channel, only updated a pixel’s value if it differed by a set threshold more than the pixels in its neighbourhood. If the pixel did not meet the threshold, the kernel size was incremented by two, up to a maximum kernel size.

A diagram of a medical scan

Description automatically generated with medium confidence

Fig. 2 - Comparison between my adaptive median filter and OpenCV's MedianBlur()

### Although the AMF was less effective in eliminating salt-and-pepper noise, both visually and quantitatively, compared to MedianBlur(), it demonstrated superior preservation of image details. When integrated into the image processing workflow, the overall image accuracy improved by 9% compared to MedianBlur().

### 2.2 Projective Transforms

All images in the dataset exhibited warping, manifesting solely as a change in perspective while retaining overall content integrity. To rectify this distortion consistently across all images, a projective transform was applied uniformly.

A close-up of a computer generated image

Description automatically generated

Fig. 3 - Analysis of Perspective Transform

Post application of the Adaptive Median Filter (AMF), the classifier achieved an accuracy of 0.69. However, upon subsequent application of the projective transform, the accuracy regressed to 0.52. Interestingly, refraining from employing the projective transform yielded an improved overall accuracy of 0.73, marking a 20% increase compared to not employing the transform.

It is noteworthy that the absence of the perspective transform not only influences the classifier accuracy but also impacts human perception of the images. Specifically, without the transform, the presence of black artifacts blending into the image edges introduces a perceptual ambiguity, potentially leading to vital information loss for medical professionals. Furthermore, in certain instances, the black artifacts assumed a blue hue, further complicating the interpretability of the images.

### 2.3 Canny Edge Detection

A systematic approach was employed to produce the mask for inpainting. Each image was converted to grayscale, to ensure the mask’s independence from colour channel values. The top-right quadrant of the image was selected as the region of interest (ROI) due to all masks falling within this area. This selection reduced any errors found from larger contours within the image, specifically from the base of the lung, affecting the produced mask.

Canny edge detection was subsequently applied to the identified region to identify areas with high gradient magnitudes, indicating edges. Employing high threshold values helped suppress noise and accentuate relevant edge information. The detected edges were transformed into contours, with the largest contour, assumed to encapsulate the missing region, selected for further analysis. A circular boundary slightly larger than the selected contour was delineated to ensure comprehensive coverage of the area requiring inpainting.

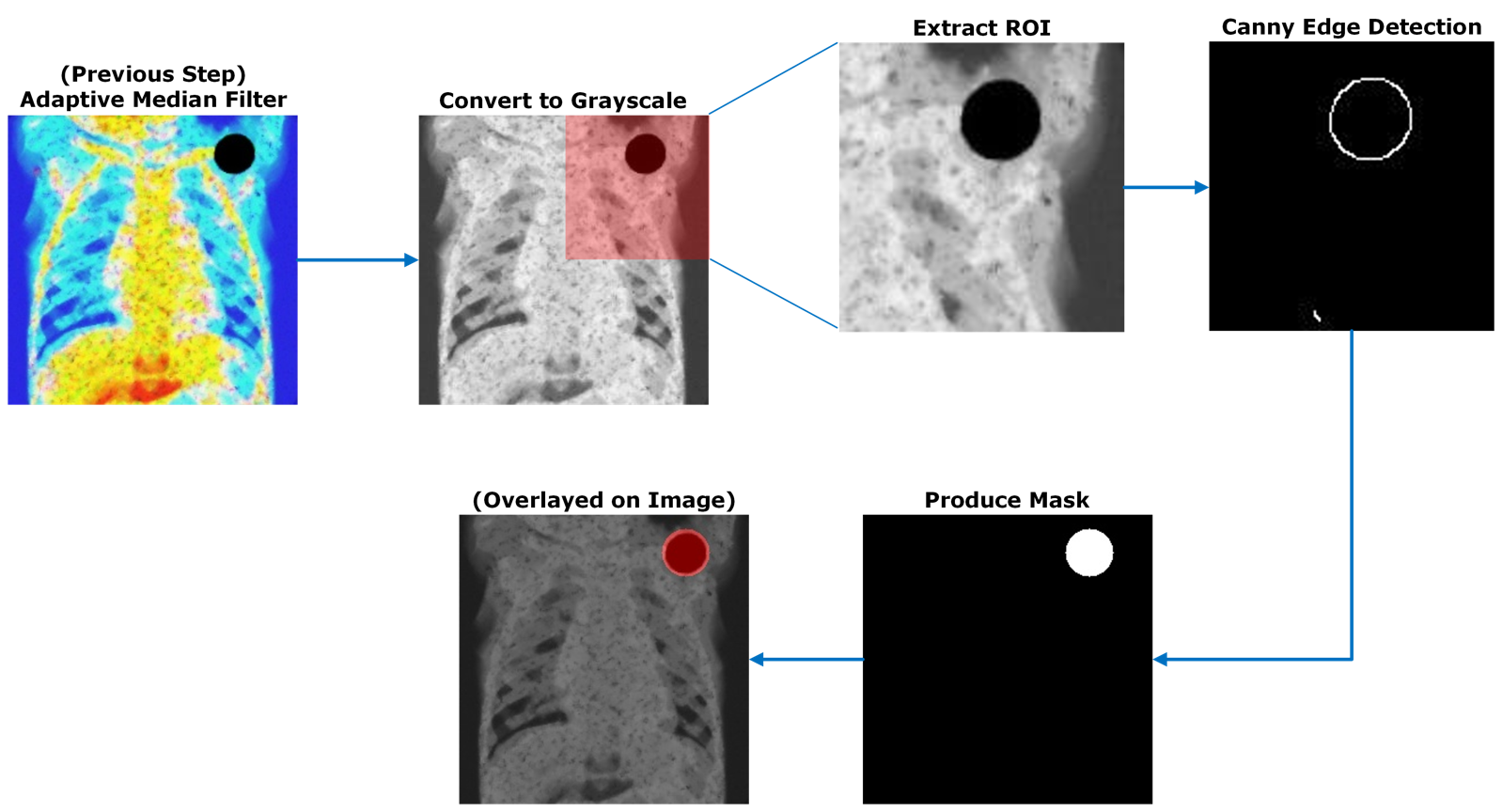


Fig. 4 – Workflow of detecting mask for inpainting

The selection of thresholds for Canny edge detection was made to effectively mitigate noise while preserving the integrity of the region to the greatest extent possible.

A black background with white circles

Description automatically generated

Fig. 5 – Edges detected between different thresholds using Canny() on im001-healthy.jpg. As can be seen, 400-500 reduces noise just enough, without removing too much of the main circle, allowing for an accurate region to be drawn.

This method has produced exemplary results.

A close-up of a radiograph

Description automatically generated

Fig. 6 – Overlaying the detected mask over greyscale images to show accuracy

### 2.4 Inpainting

Inpainting was achieved using a slightly adapted version of a python implementation [1] of the region filling algorithm proposed by Criminisi et. al. [2]. The algorithm uses exemplar-based image inpainting by using the isophote information of boundary pixels to extend linear structures. The algorithm also extends around the initial patch in a clockwise motion, improving linear structures compared to traditional concentric approaches.

As can be seen

### 2.5 Non-Local Means Denoising

Despite the AMF removing much of the salt-and-pepper noise, significant amounts of noise remained. To combat this, I employed a Non-Local Means (NLM) Denoiser, through OpenCV’s fastNlMeansDenoisingColored() function. NLM works by dividing the image into patches, and comparing the similarity of patches, before weighting these. Noise is then removed from patches by replacing pixels with the weighted average of similar patches.

I used NLM twice throughout the processing workflow, in order to gradually remove noise whilst preserving details, as well as removing noise introduced from other methods.

The first NLM instance used a h and hColor of 12, with a templateWindowSize of 7, and searchWindowSize of 21, and is introduced first after the inpainting.

A close-up of several images of a person's body

Description automatically generated

As you can see from Fig.X, NLM effectively removed almost all noise, at the expense of a loss of clarity. Colours are more muted, and edges not well defined. That being said, the image is qualitively improved. The final repaired image with the first NLM present is only 1% more accurate than without, however not including the first NLM leads to the remaining salt-and-pepper noise being amplified, losing any benefits of increased clarity.

### 2.6 Laplace Transform

NLM had removed clarity and contrast from the image. In order to regain this, I used a Laplace transform. Using a kernel of size 3, I detected regions of rapid intensity change, and subtracted this from the original image. This highlights edges, improving overall visual appeal, with the slight introduction of noise.

### 2.7 Histogram Equalisation in the HSV Colour Space

The damaged images appear to have a colour distribution not matching those of the reference images. Also, some colour channels appear brighter than others. My first step to improve this is through Histogram Equalisation. I completed Histogram Equalisation in the HSV colour space, specifically the Variance channel, as this was able to update the contrast and brightness of an image, without affecting individual colour channels.

### 2.8 Reapplication of Non-Local Means Denoising

### 2.9 Contrast Limited Adaptive Histogram Equalisation in the LAB Colour Space

### 2.10 Sobel Edge Enhancement

### 2.11 Colour Scaling

## 3.0 Analysis

### 3.1 Analysis of Classifier Accuracy

### 3.2 Analysis of Image Quality

# Bibliography

|  |  |
| --- | --- |
| [1] | N. Nahar, “GitHub Repository,” 11 12 2021. [Online]. Available: https://github.com/NazminJuli/Criminisi-Inpainting. |