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

**James Duxbury (bdzz75)**

## **Overview**

I was provided with a set of 100 coloured X-Ray images of patient’s lungs. These images suffered with significant distortions, and I was tasked with processing these images to improve the classifier’s accuracy.

The classifier had an accuracy of 0.55 before any processing.

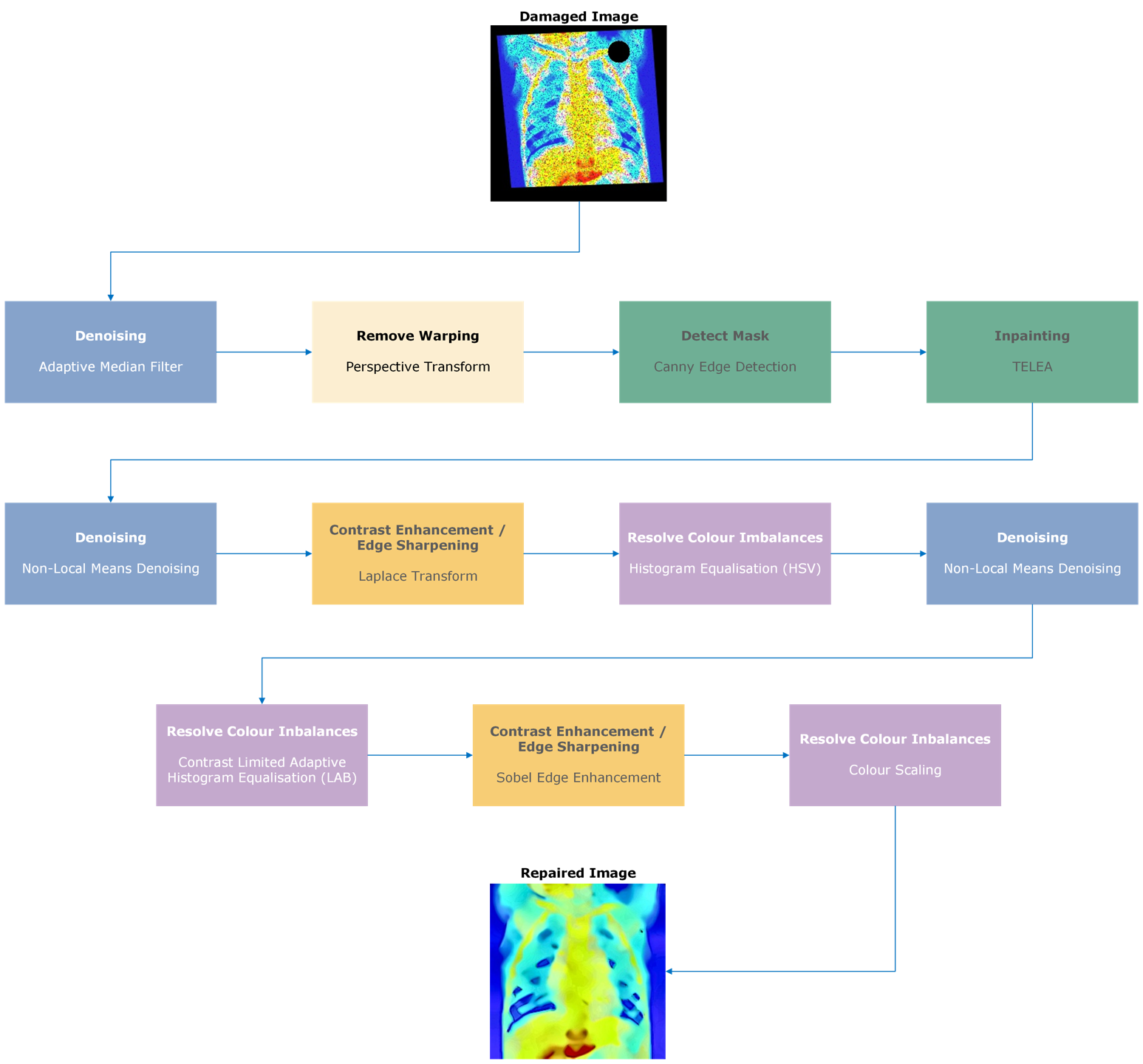
Analysis includes the accuracy of the classifier with and without the step, as well as individual performance.

Fig. - Overview of Image Processing Process

## 2.0 Image Processing

### 2.1 Adaptive Median Filter (AMF)

All images contained significant amounts of Gaussian and salt-and-pepper noise, which posed issues as I had to strike a balance between retaining image details and removing noise.

A diagram of a medical scan

Description automatically generated with medium confidenceSalt-and-pepper noise is best removed with a median filter, however during testing, I found OpenCV’s MedianBlur() function to cause too much blurring and loss of details. Therefore, I implemented an Adaptive Median Filter (AMF) function that, for each colour channel, produced a neighbourhood and calculated the median value. If the pixel’s value was over a set threshold (in my case 10), then the pixel was replaced with that of the neighbourhood. If the pixel was within the threshold, then the neighbourhood kernel size was increased by two, up to a set maximum kernel size.

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

My AMF was less efficient at removing salt-and-pepper noise alone than MedianBlur(), however it preserved image details at a much better rate. Combined with my further denoising and image processing workflow, the overall image accuracy was 9% greater.

### 2.2 Projective Transforms

All images were warped, affecting only their perspective. All images had the same warping, therefore I applied a projective transform.

The images had a classifier accuracy of 0.69 following the AMF, however after applying the perspective transform, accuracy had dropped to 0.52. That being said, processing the image without a perspective transform led to an overall accuracy of 0.73, which is 20% lower than with the perspective transform in place. No transform also impacts human perception of the images – the black blends into the edges of the images, which can potentially lead to a loss of information that may be vital to doctors. In some cases, the black was made blue.

A close-up of a computer generated image

Description automatically generated

Fig. 3 - Analysis of Perspective Transform

### 2.3 Canny Edge Detection

Next, I worked on detecting the mask for inpainting the missing region. Each image has regions of different sizes and positionings, so I used the below procedure to extract the mask.

First, a copy of the image is converted to greyscale, as the mask is independent to any colour channel values. I then extract the top right of the image as the region of interest, this being because the missing regions are exclusively in the top-right quadrant, and by isolating this area, larger contours (especially created by the lungs themselves) are not impacting the detection of the mask. I then use Canny edge detection to identify areas of high gradient magnitude, indicative of edges, with high thresholds to reduce noise. These are then transformed into contours. I select the largest contour, which will be the missing region, and enclose this with a circle that is slightly larger, to ensure no area is missing. A mask is then produced using this circle, and converted to colour for inpainting.

A close-up of a radiograph

Description automatically generated

Fig. 4 – Workflow of detecting mask for inpainting

A black background with white circles

Description automatically generatedThe choice of thresholds for Canny edge detection was chosen as it removed a lot of noise, whilst maintaining as much of the region as possible.

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 perfect results for the x-ray images provided.

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 OpenCV’s inpaint() function, which is set to use the algorithm of Telea [REF].

It was chosen to use OpenCV’s method due to the fact that a previously implemented version of Crimini’s inpainting algorithm [REF] led to a runtime of around 5 hours, and actually produced worse accuracy with the classifier. Although the inpainting itself is not fully accurate, differences become negligible once further NLM denoising and further processing completes, producing

### 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

### 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