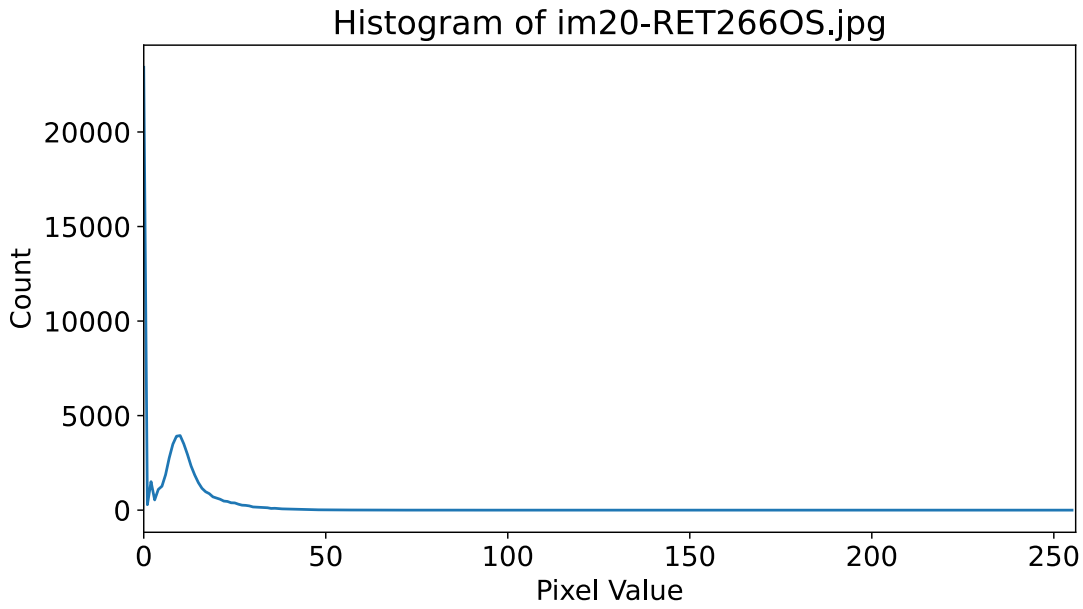
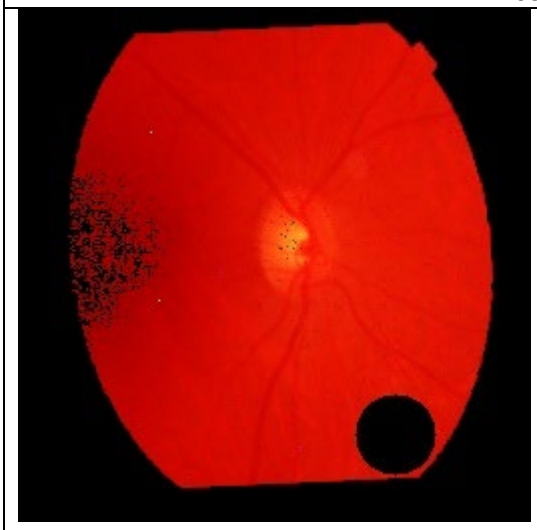
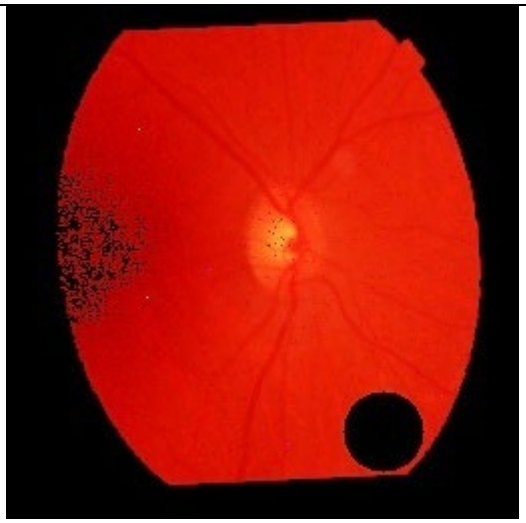
Image Processing Report

Brightness and Contrast Adjustment

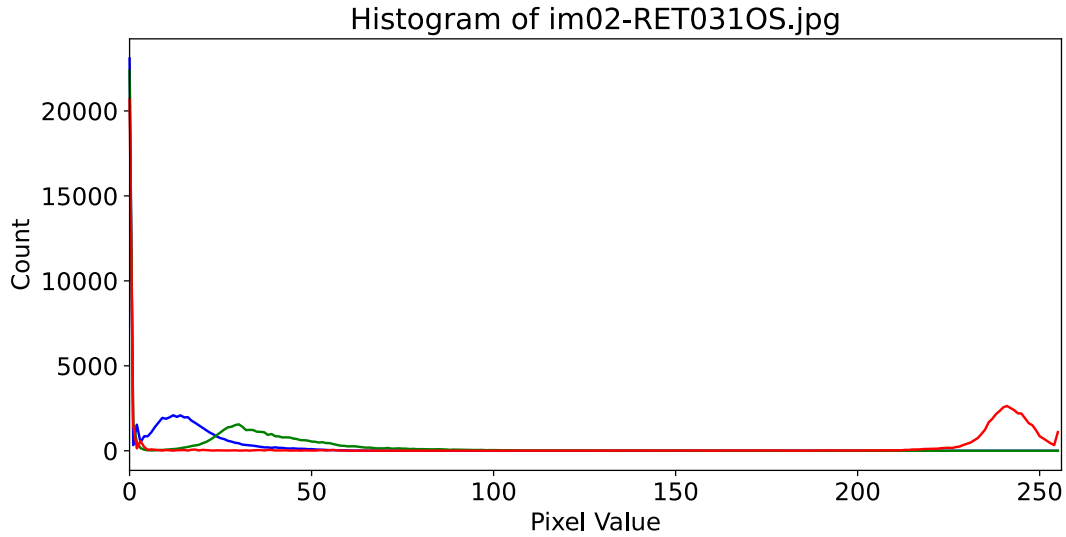
From observing the images and their histograms, it was evident that they lacked contrast. Additionally, many of them were underexposed, with a large spike at low intensities and sometimes a second peak slightly to the right.



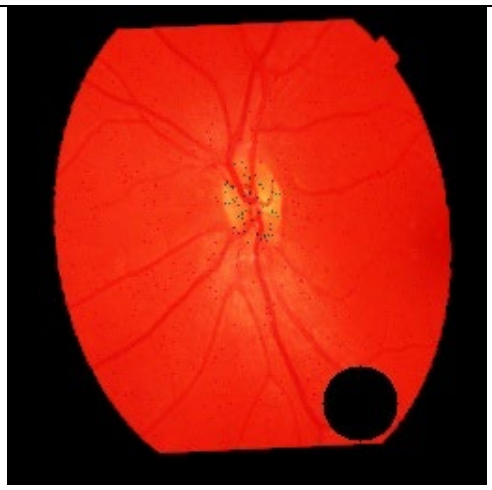
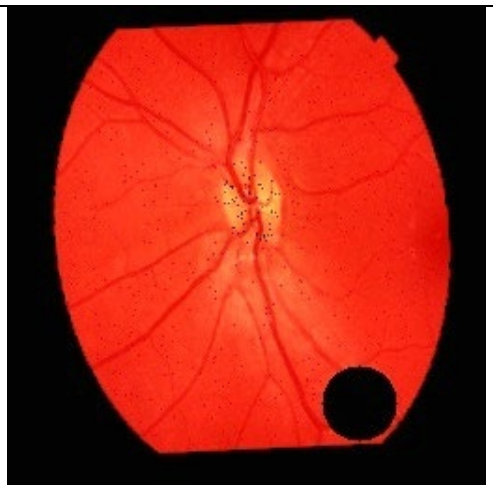
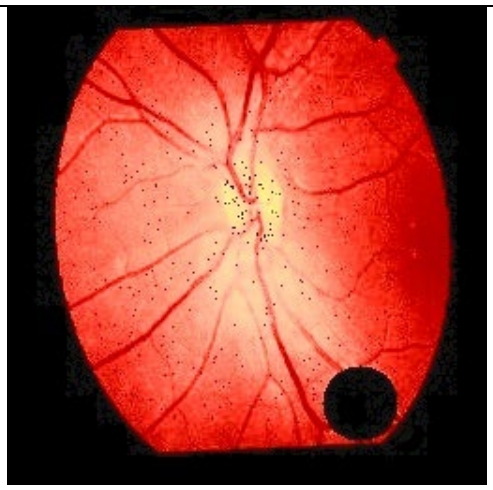
Applying gamma correction seemed a suitable approach to remove this underexposure, so I applied it with a gamma of less than 1. In addition to correcting the brightness, it also meant some of the black noise in the images was reduced.

im03-RET033OD	
	
Original	Gamma corrected image, Gamma = 0.8

From there, I investigated the brightness of the different colour channels. The blue and green channels had a small dynamic range, however, the red channel was more complex. It had a large dynamic range, with both high and low intensity values with few values in between. What was most notable, was how narrow the peaks for all the channels were, indicating the use of a method to spread them out.



I considered several methods to increase contrast. Due to the vast amount of salt and pepper noise, an approach such as contrast stretching would only be useful if it was robust to outliers. Instead, I experimented with equalization techniques, but could not produce a good performance. Upon further research, this was due to the images having a large area, low intensity background, and so an adaptive approach was suggested.[1] I therefore, tested this using CLAHE and found it helped produce high quality, well performing images.

im02-RET031OS		
		
Original	CLAHE: cliplimit=1, tileGridSize= (8,8)	Standard Histogram Equalization

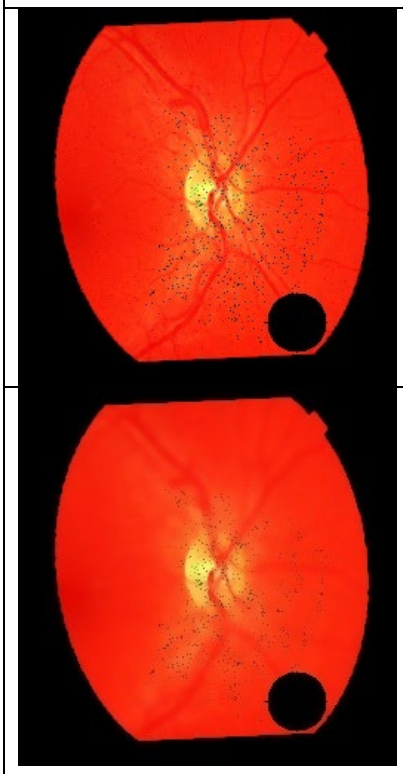
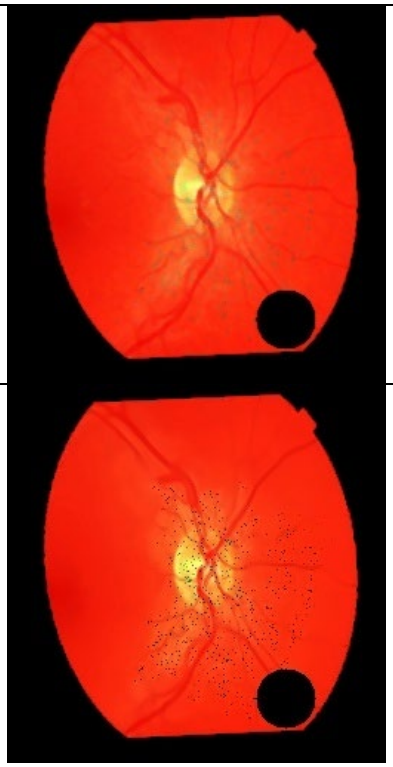
[1] Toet, A., & Wu, T. (2014). Efficient contrast enhancement through log-power histogram modification. Journal of Electronic Imaging, 23, 063017. https://doi.org/10.1117/1.JEI.23.6.063017

When using CLAHE, I discovered that I could alter performance significantly through the size of the tiles, finding 3 to be a good value.

Noise Reduction

Through observing the images, the largest source of noise appeared to be salt and pepper. I, therefore, directed my approach to primarily focus on removing this.

It is known that median based filter methods perform well at removing salt and pepper noise, so this is where I started experimenting. I also attempted some advanced noise removal techniques such as applying a bilateral filter and a non-local means, yet they performed worse. The bilateral filter, although edge preserving, also preserved much of the salt and pepper. For the non-local means it either did not remove enough noise or too much, in which the image appeared smooth looking.

<table>
<tr><td colspan="4">im04-RET037OD</td></tr>
<tr><td rowspan="2">  </td><td>Original Image</td><td rowspan="2">  </td><td>Median Filter: kernelsize= 3</td></tr>
<tr><td>Billateral Filter:

d= 9
sigmaColor = 75
sigmaSpace = 75</td><td>Non-local means denoising:

h_luminance =4
photo_render =4
search_window = 7
block_size = 21</td></tr>
</table>

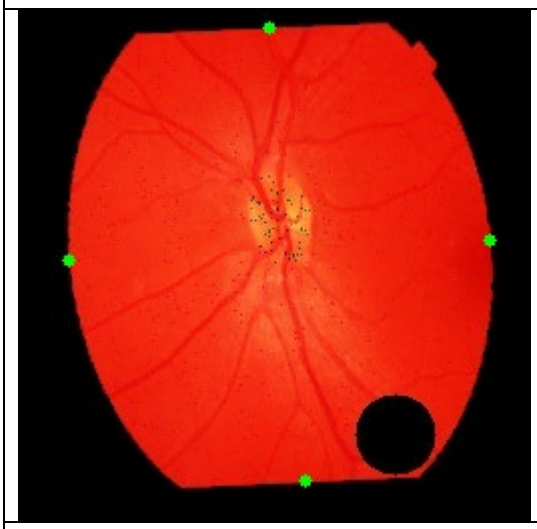
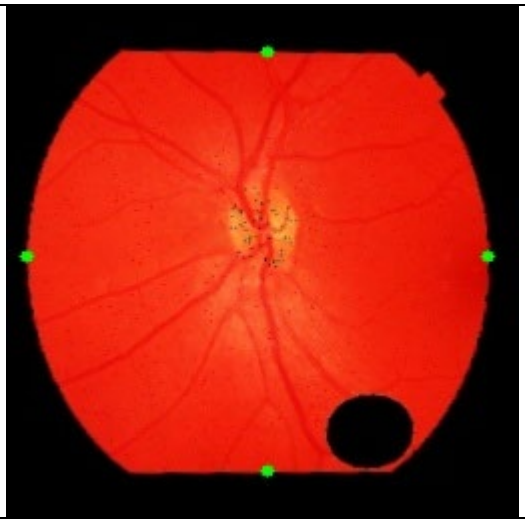
Once using the median filter, I experimented with kernel sizes, finding that a kernel of 3 x 3 worked best. Any more than that started to obfuscate some of the finer detail.

To further reduce noise, I devised an algorithm to lessen the effect of unwanted black pixels in the image. The algorithm first applies thresholding to create a mask of pixels darker than a certain value and then applies inpainting to the pixels withinside this mask. Although this method didn't improve performance when applied independently, I found it did when used in tandem with CLAHE. Additionally, visually some of the images look better when it is applied, and accuracy was increased.

Perspective Adjustment

As the images had been warped, it was necessary to apply transformations to make them more realistic. I noticed that the images were not circular in shape and rotated slightly.

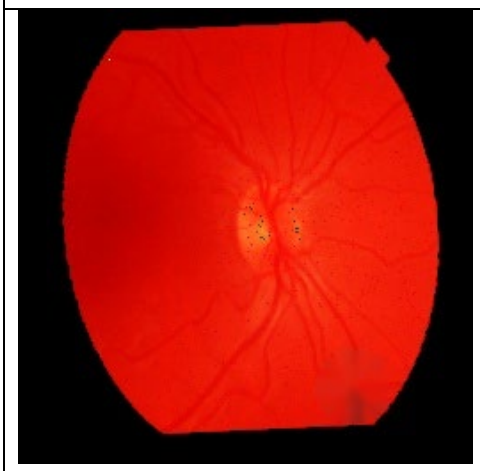
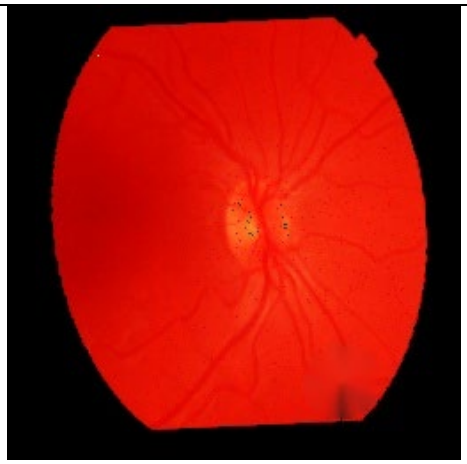
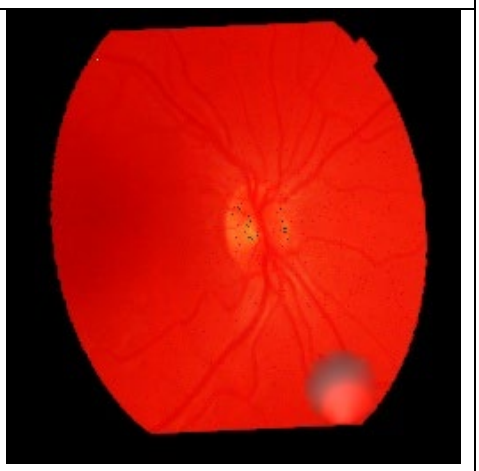
To remedy this, I applied the cv2 warp perspective function to each image. In doing so, I supplied a set of source and destination points. The source points were fit around the eye such that they were the points intersected by the minor and major axis. The destination points were set such that they were horizontally and vertically aligned, and that the size of the ellipse created, would be more circular. By doing this, I produced a more circular shape and removed the rotation.

im02-RET031OS	
	
Original	Perspective Warped

It was interesting to discover how much impact the source and destination points had on the performance of the model, with tiny pixel variations inducing drastically differing accuracy.

Inpainting

To fill in the large gap in the image, I explored several inpainting methods and parameters. Firstly, were two methods provided by OpenCV. Although I found little visual difference between them, they seemed to affect the accuracy of the model still somewhat. Furthermore, I experimented with scikit-image's biharmonic inpainting method, but this produced lower quality results. Ultimately, the Navier-Stokes based method performed the best, so I opted to use that.

im01-RET029OD		
		
INPAINT_NS Inpaint Radius = 3	INPAINT_TELEA Inpaint Radius = 3	Biharmonic inpaint