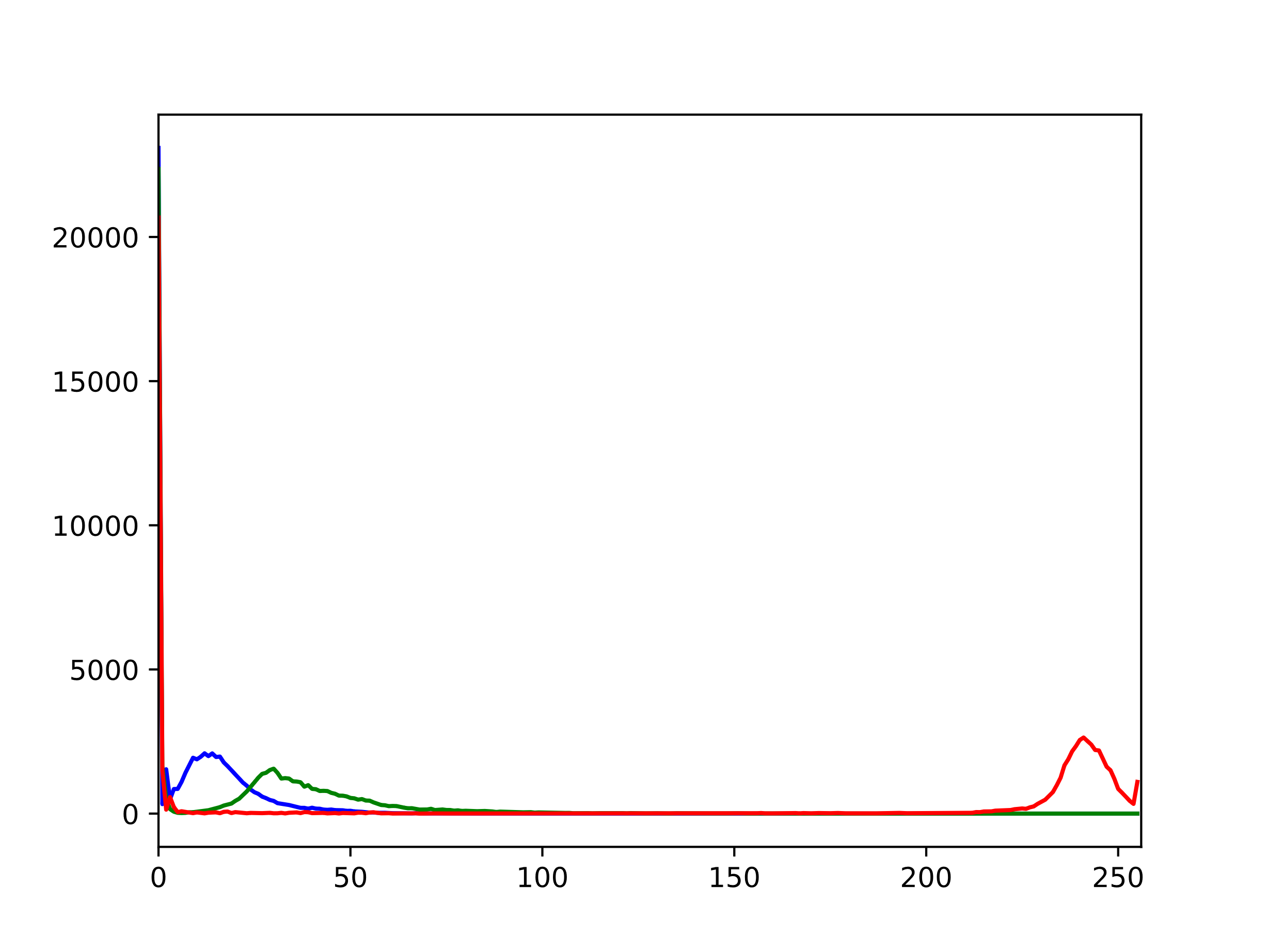
Image Processing Report

# Introduction

* Pipeline used was: gamma correction -> custom salt and pepper technique -> clahe -> median filter -> perspective warping

# Brightness and Contrast Adjustment

* When looking at a histogram of the brightness channel, it backed up the clear lack of contrast. Many also appeared underexposed, with the peak of the histogram to the left of 125 and large spike for dark values.
* (DIAGRAM WITH HISTOGRAM OF BRIGHTNESS)
* Applied gamma correction, to increase performance
  + Due to underexposure applied gamma < 1. Helped to reduce black pepper noise in images, and noise reduction further down the image processing pipeline
* Checked histograms for individual colour channels
  + Red channel is far brighter than green and blue
  + Blue and green Histograms have small dynamic range with exception to red. Red is complex, with both high and low values and few mid values.
  + Very little contrast and little spread. Could benefit from equalization to be able to discern features better.
* Considered different methods to improve contrast
  + Lots of salt and pepper noise, so contrast stretching would only help if using a method that was robust to outliers
  + Experimented with histogram equalization techniques. Due to small dynamic range, (in all channels except red) it did not appear as good. Poor results with classifier also.
  + Instead applied CLAHE. Experimented with tile size and made large difference in performance  (TWO IMAGES, SIDE BY SIDE OF TILE SIZE PERFORMANCE). To do this image converted to ycrb and applied to the y channel. Then converted back.

# Noise Reduction

* From observing images largest source of noise appeared to be salt and pepper. I therefore focused primarily on using a filter to remove this, and the other noise in the image.
  + Best method to remove salt and pepper is median filter
  + Attempted advanced techniques like bilateral filter also to preserve some of the fine edges, however, preserved much of the salt and pepper also.
  + Attempted non-local means with a variety of parameters. Low values did not remove the salt and pepper noise enough, whilst high values made the images too smooth looking. Consequently, performance suffered.
  + Most effective found was local means with a kernel size of 3. The larger kernels removed to much of the images finer detail.
  + (SHOW DIAGRAM WITH IMAGE OF BILLATERAL, NON-LOCAL MEANS AND MEAN FILTER)
* Tried my own method at removing salt and pepper
  + I noticed that black pixels withinside the main body of the image were never natural. Always noise
  + As a result, I wrote a method that applies thresholding to create a mask of pixels darker than a certain amount. I then apply inpainting to the pixels withinside this mask, to try and lessen their effect in the image.
  + Although this method I wrote doesn’t affect performance when applied exclusively, it does in tandem with clahe .Additionally, visually some of the images look better when it is applied (particularly with black noise at the edge of the image) (SHOW TWO IMAGES, ONE APPLYING THIS METHOD, AND ONE NOT

# Perspective Adjustment

* Can visually see that the images are not circular shaped, and rotated at a slight angle.
* Compared them to the original images, to see how exactly they had been adjusted.
* As only the perspective had been warped, applied the perspective warp cv2 function
  + To do this had to supply source and destination points
  + Manually adjusted some source points to fit around the eye such that there were two sets of points, on opposite sides to one another.
  + Destination points were two set of points that pass through a horizontal and vertical line respectively. Process of trial and error to produce a shape that looked realistic. Noticed that the model was very sensitive to the shape, and even a pixel difference, caused the model to classify differently
  + (SHOW IMAGE OF GREEN POINTS ON EYE IMAGE) then show corrected version with green points

# Inpainting

* Experimented with several ways of inpainting. I used both methods provided by opencv, which are INPAINT\_NS and (WHAT WAS IT?). Additionally, tried out sckit-images biharmonic inpainting method.
* Found mild difference between them
* (Diagram showing three inpaint methods)