## Contents

# 1. Introduction

## 1.1 Background

Histogram equalization (HE) is a technique in the field of image processing that aims to enhance the visual quality of digital images. This method improves the contrast and brightness of a given image, making it more visually appealing and easier to analyze.

In this report, we will be implementing the algorithm by following the steps outlined in section 1.2. We will apply this method on 8 sample images, then discuss the pros and cons of this algorithm with reference to the augmentation done to them. Lastly, we will discuss and implement improvements, in particular, Contrast-Limited Adaptive Histogram Equalization, to obtain better output images.

## 1.2 Theory

In a given RGB image, each pixel has three values associated to the R, G and B channels. The pixel values indicate the intensity of each of the channel. In a low contrast image, the pixel values in most of the pixels (in all three channels) are very similar. Visually, the histograms for the pixel values are clustered around a single region (Figure 1).

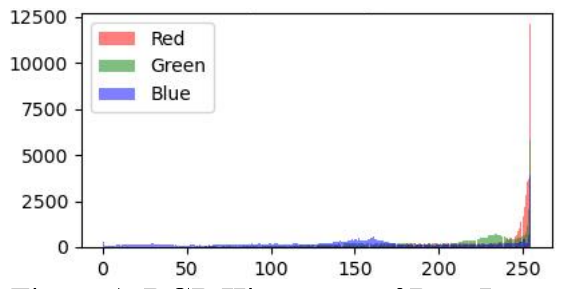


**Figure 1: RGB Histogram of Raw Image**

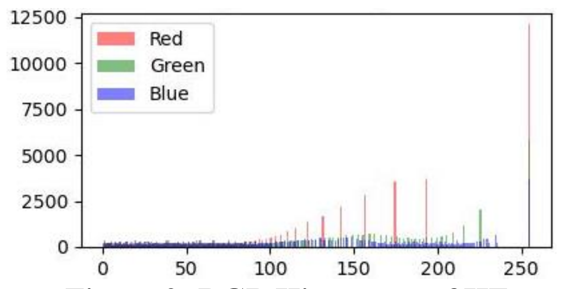


**Figure 2: RGB Histogram of HE-Processed Image**

Therefore, to increase the contrast of an image, we hope to distribute pixel values more evenly, so that the histograms do not cluster as much (Figure 2).

Let the width and height of an image be $W$ and $H$, then the total number of pixels in that image, $N = W \times H$. Furthermore, let $L$ be the total possible pixel values of a histogram.

Let the number of pixel (in the original image) with pixel value $j$ be $N_j$. Then, the probability mass function of the original image is given to be

$$p_{original}(x) = \frac{N_x}{W \times H}$$

the cumulative distribution function (CDF) of the pixel values in the original image is:

$$\begin{aligned} c_{original}(x) &= p(X \leq x) \\ &= \frac{\sum_{j=1}^{x} N_j}{W \times H} \end{aligned}$$

In the ideal scenario, we would like the CDF of the pixel values to be,

$$c_{normalized}(x) = \frac{x}{L-1},$$

as the pixel values would be uniformly distribution in the equalized image.

Suppose a pixel $P$ has pixel value $k$ in the original image and has a pixel value $k'$ after the equalization. To find the $k'$, we have

$$\begin{aligned} c_{normalized}(k') &= c_{original}(k) \\ \frac{k'}{L-1} &= \frac{\sum_{j=1}^{k} N_j}{W \times H} \\ k' &= (L-1) \times \frac{\sum_{j=1}^{k} N_j}{W \times H} \end{aligned}$$

The intuition here is that pixels that are less bright than pixel $P$ in the original image, should still be less bright than pixel $P$ in the equalized image. This preserves the relative brightness of pixels but improving the contrast.

Therefore, we can systematically find the pixel values of all the pixel in the equalized image, using the CDF obtained from the original image, thereby producing the equalized image.

## 2. Implementation and Discussion of HE

In this section, we will implement the HE algorithm on 8 sample images, on each of the channel independently. The algorithm will produce both grayscale and RGB images. Then, we will discuss the results we obtained from the implementation.

### 2.1 Explanation on Implementation of HE Algorithm

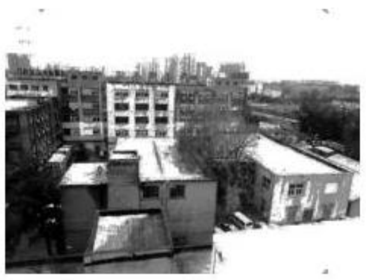
The following are the steps we will be taking to obtain the equalized image from the raw RGB images.

1. <u>Processing the Images and Obtaining the Pixel Values of Each Channel</u>
    1) Load the images using the *read_jpeg_files_in_folder* function.
    2) Use the *Pillow* module to convert the RGB images into grayscale images.
    3) Additional to step 2, we will split the image into its R, G and B channels.
    4) Obtain a sequence of *ph_list*, one for each image, where *ph_list* contains the grayscale, red, green, and blue pixel values, respectively.
2. <u>Obtaining the Equalized Grayscale and RGB Images</u>
    1) For each image, we obtain the histogram of each of the channel (including grayscale), using *np.histogram.*
    2) Next, we find the CDF of each channel by using the *histogram.cumsum()* method, and dividing it by the size of the image (i.e., $W$ and $H$)
    5) Normalize the CDF of each channel by taking the product of $(L-1)$ with the CDF. In this case, $L = 255 + 1 = 256$.
    6) Calculate the final pixel value of each channel by clipping (i.e., rounding) the normalized CDF to nearest integer.

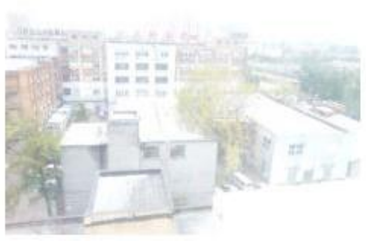
To output the grayscale images, we only have to work with the equalized grayscale image values, while to output the RGB images, we have to merge the equalized R, G, and B image values.

## 2.2 Image and Histogram Results from HE

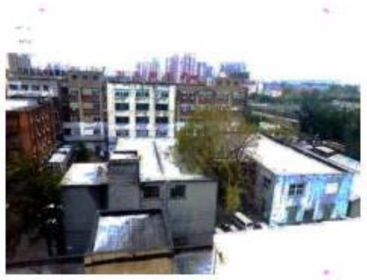
(a) Original Gray



(b) HE Gray

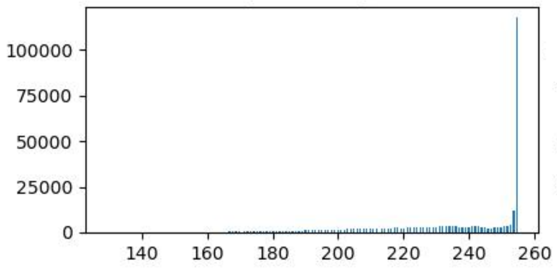


(c) Original RGB

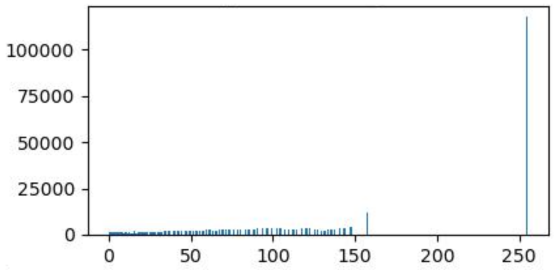


(d) HE RGB

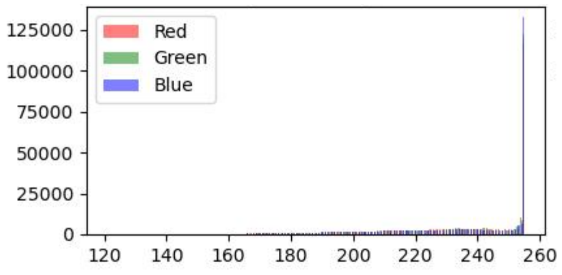
**Figure 3: Image Results from Sample 1**



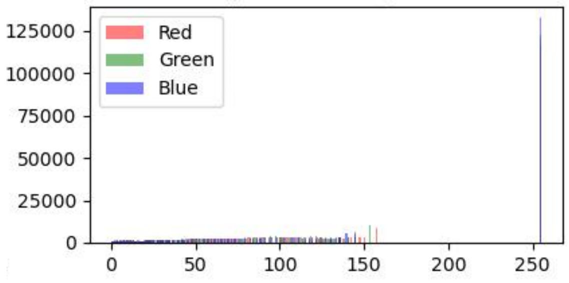
(a) Original Gray



(b) HE Gray

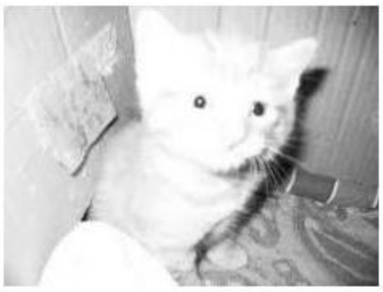


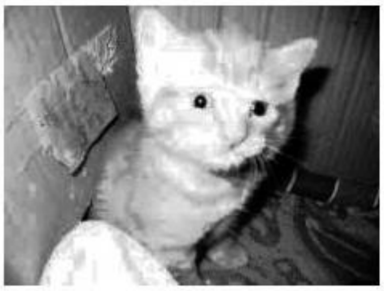
(c) Original RGB

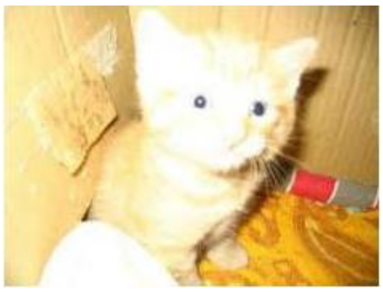


(d) HE RGB

**Figure  4: Histogram Results from Sample 1**


(a) Original Gray


(b) HE Gray


(c) Original RGB

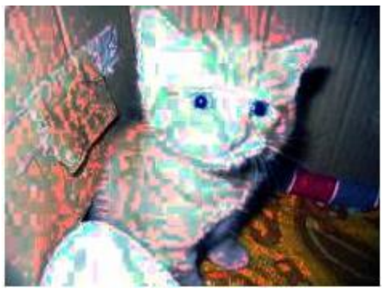
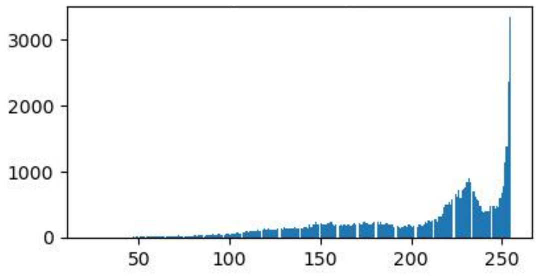
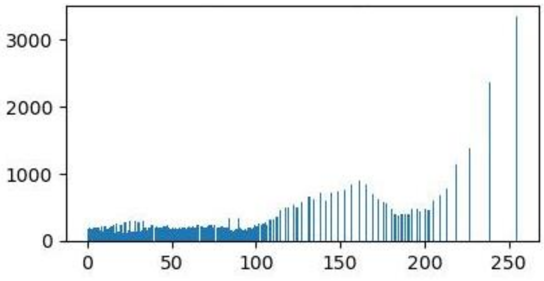

(d) HE RGB

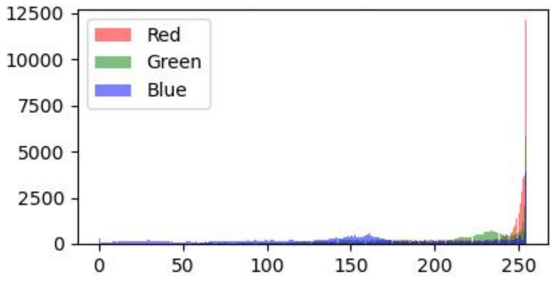
**Figure 4: Image Results from Sample 2**



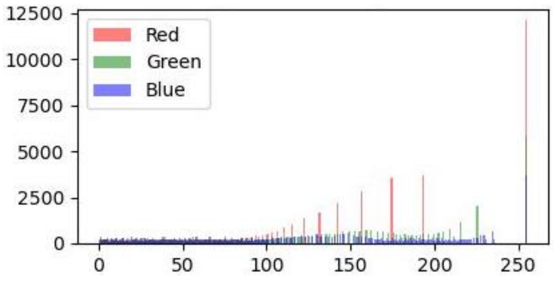
(a) Original Gray



(b) HE Gray

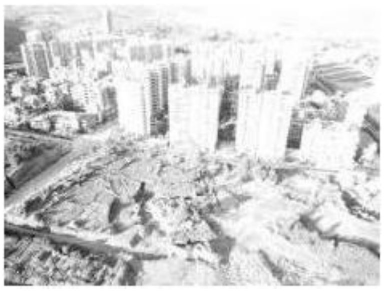


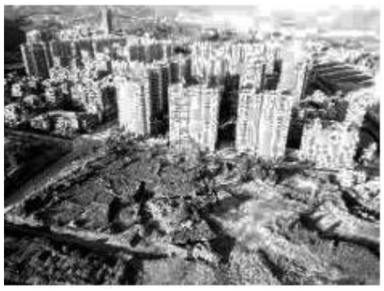
(c) Original RGB

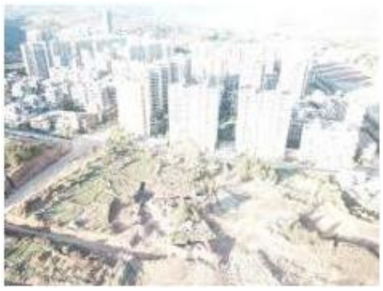


(d) HE RGB

**Figure 5: Histogram Results from Sample 2**


(a) Original Gray


(b) HE Gray


(c) Original RGB

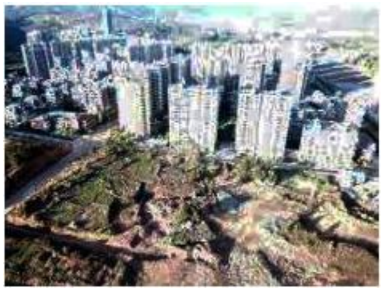
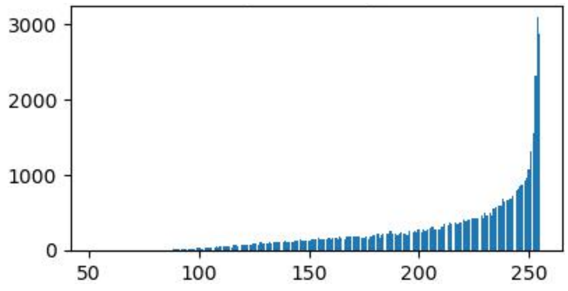
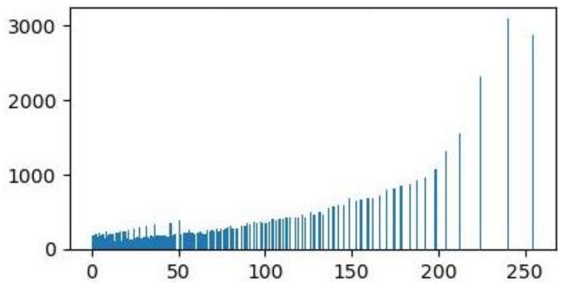

(d) HE RGB

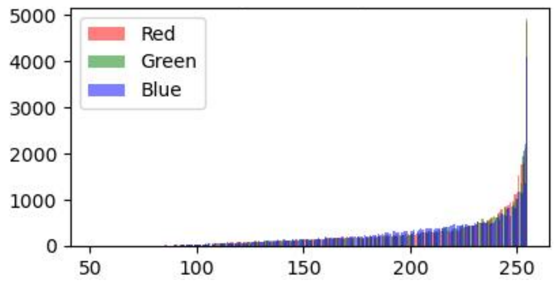
**Figure 6: Image Results from Sample 3**



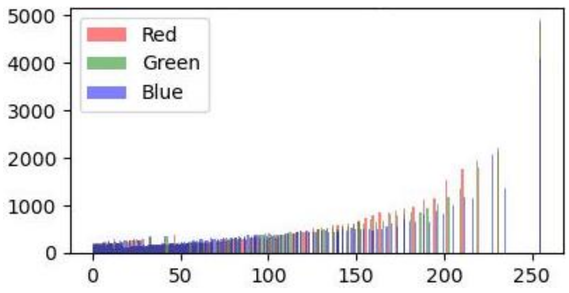
(a) Original Gray



(b) HE Gray

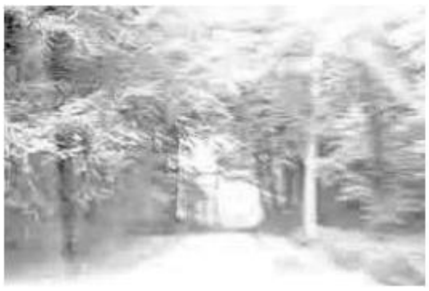


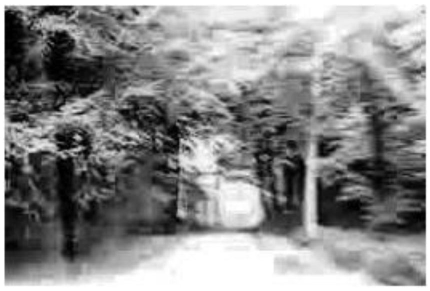
(c) Original RGB

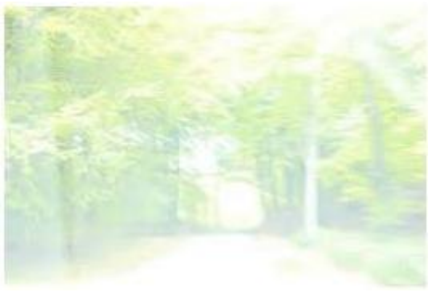


(d) HE RGB

**Figure  7: Histogram Results from Sample 3**


(a) Original Gray


(b) HE Gray


(c) Original RGB

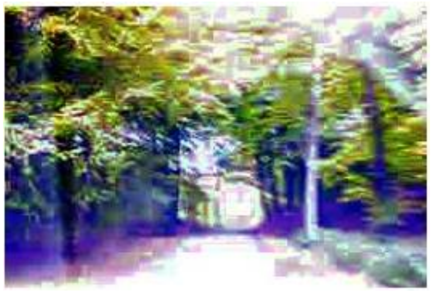
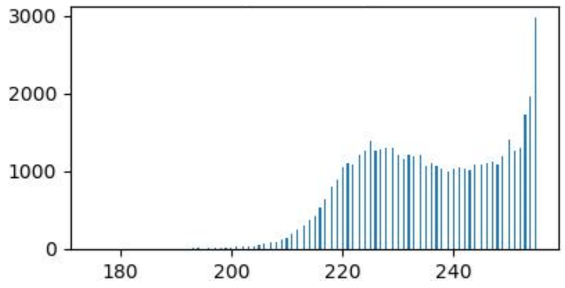
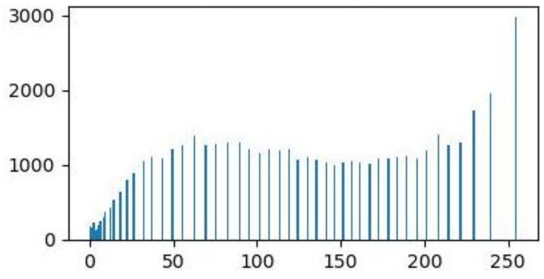

(d) HE RGB

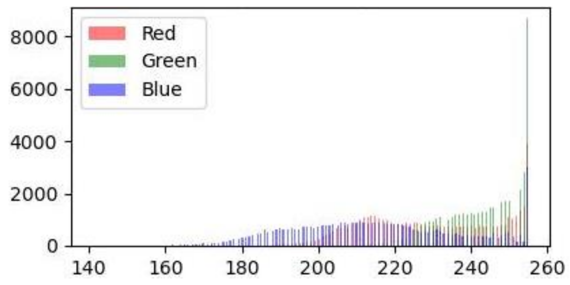
**Figure 8: Image Results from Sample 4**



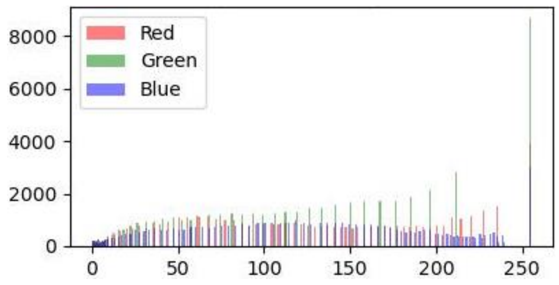
(a) Original Gray



(b) HE Gray

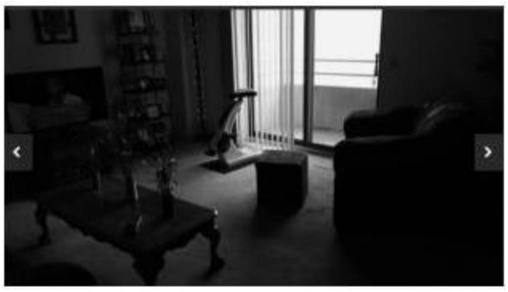


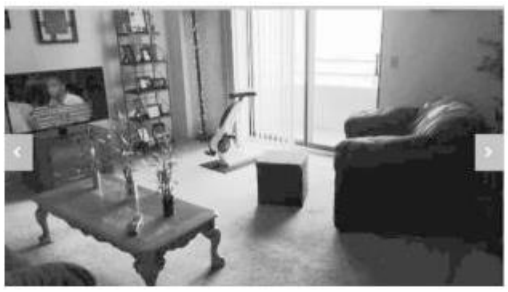
(c) Original RGB

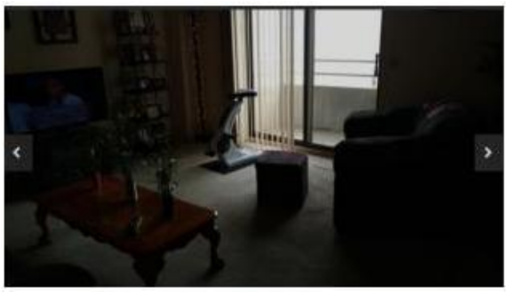


(d) HE RGB

**Figure  9: Histogram Results from Sample 4**


(a) Original Gray


(b) HE Gray


(c) Original RGB

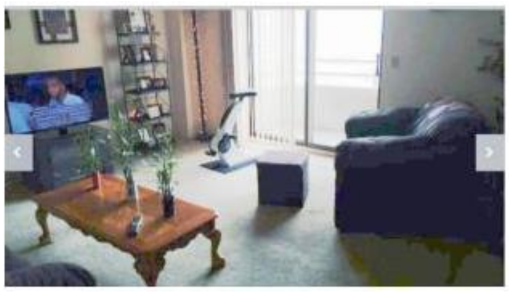
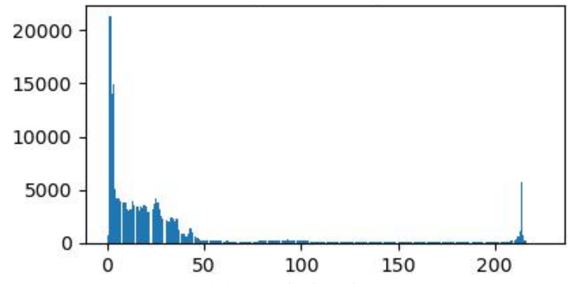
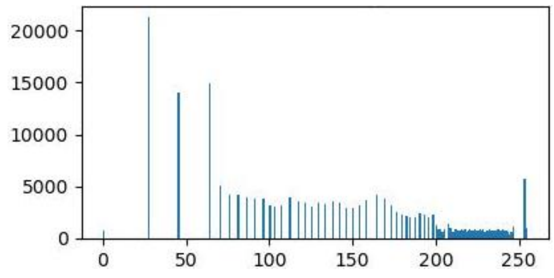

(d) HE RGB

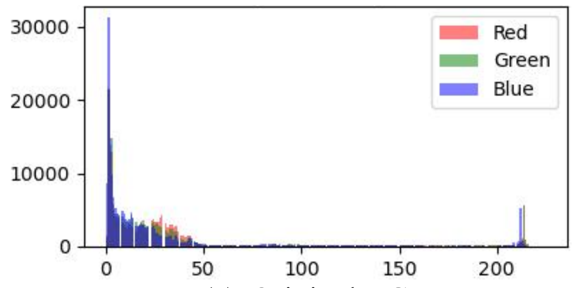
**Figure 10: Image Results from Sample 5**



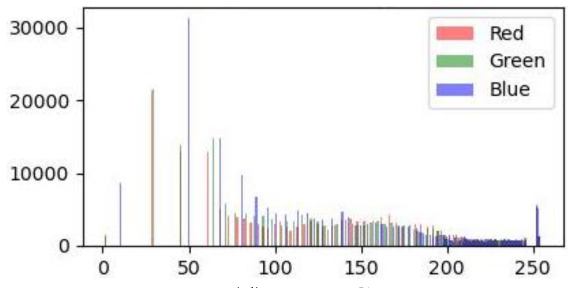
(a) Original Gray



(b) HE Gray

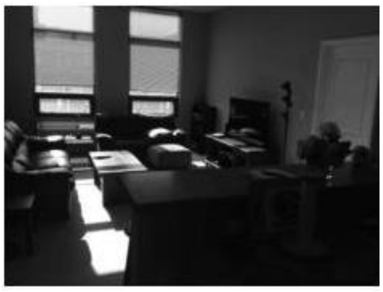


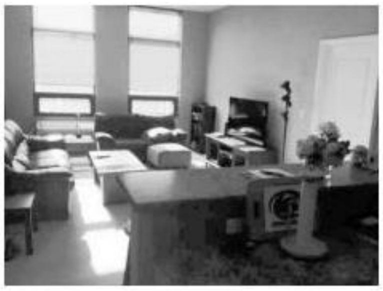
(c) Original RGB

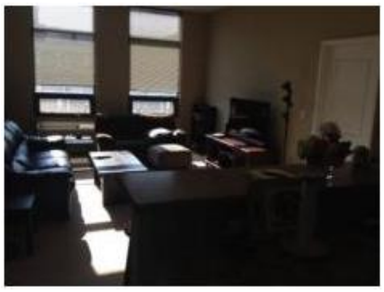


(d) HE RGB

**Figure 11: Histogram Results from Sample 5**


(a) Original Gray


(b) HE Gray


(c) Original RGB

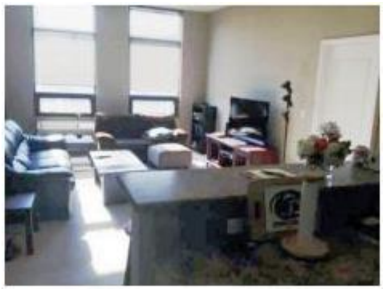
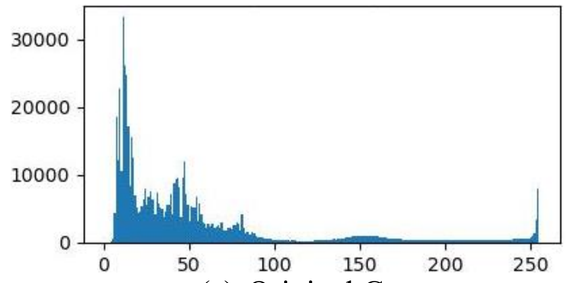
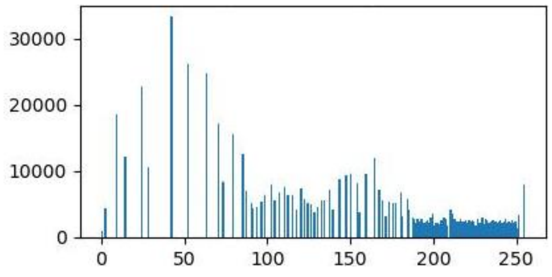

(d) HE RGB

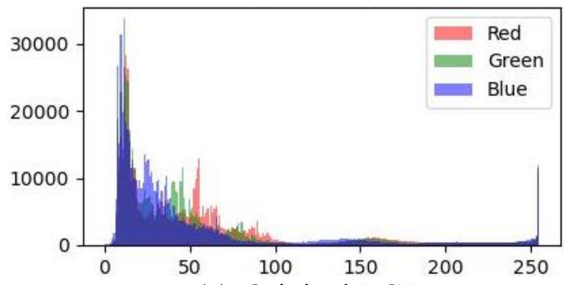
**Figure 12: Image Results from Sample 6**



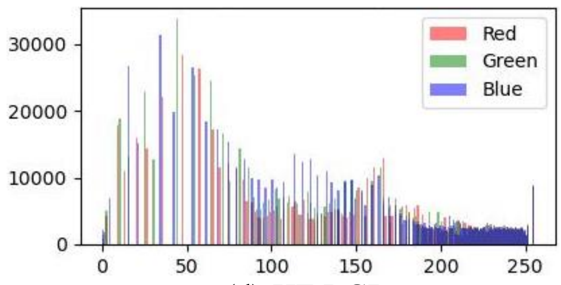
(a) Original Gray



(b) HE Gray

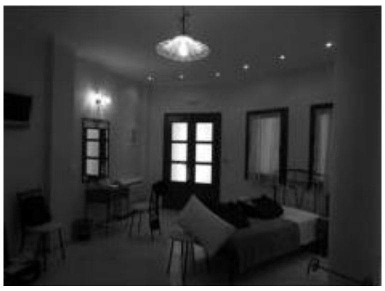


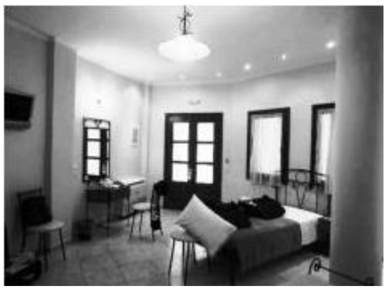
(c) Original RGB

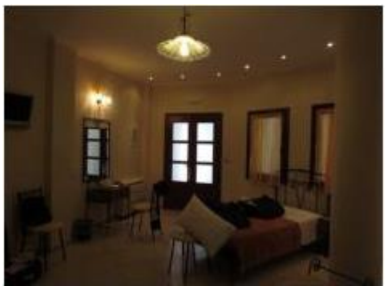


(d) HE RGB

**Figure  13: Histogram Results from Sample 6**


(a) Original Gray


(b) HE Gray


(c) Original RGB

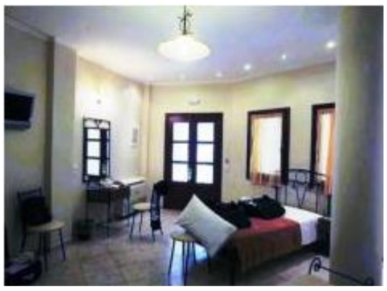
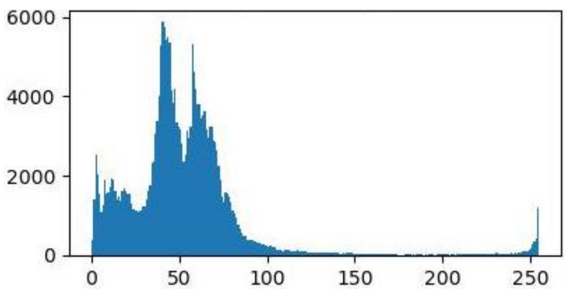
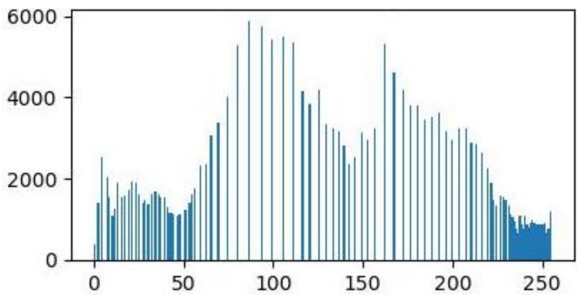

(d) HE RGB

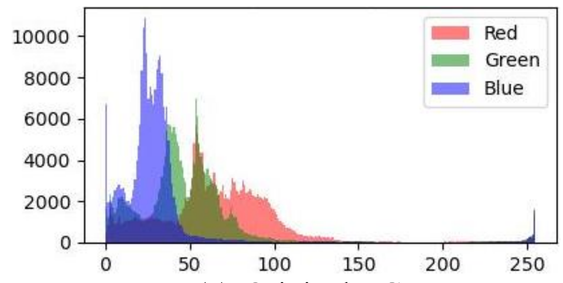
**Figure 14: Image Results from Sample 7**



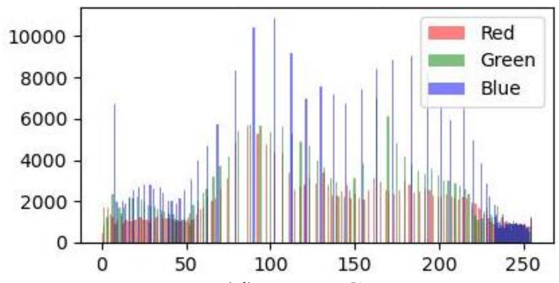
(a) Original Gray



(b) HE Gray

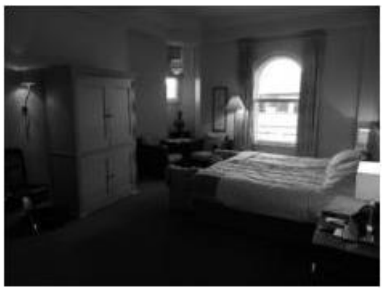


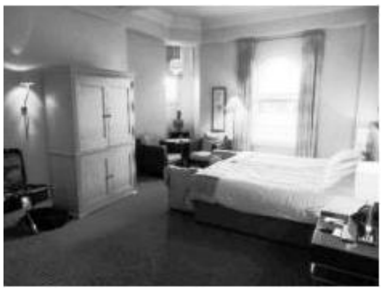
(c) Original RGB

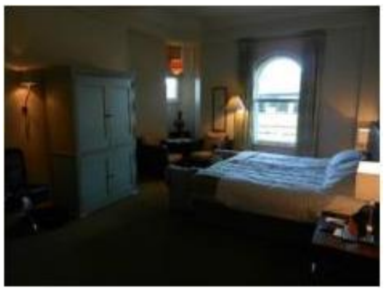


(d) HE RGB

**Figure  15: Histogram Results from Sample 7**


(a) Original Gray


(b) HE Gray


(c) Original RGB

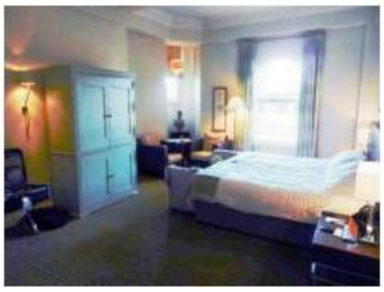
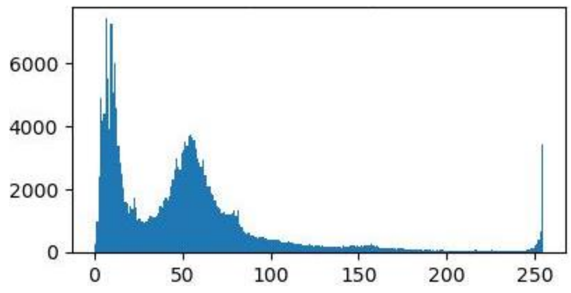
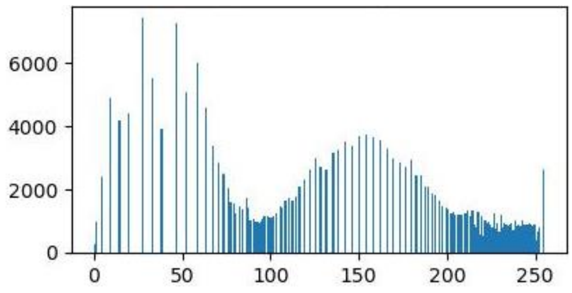
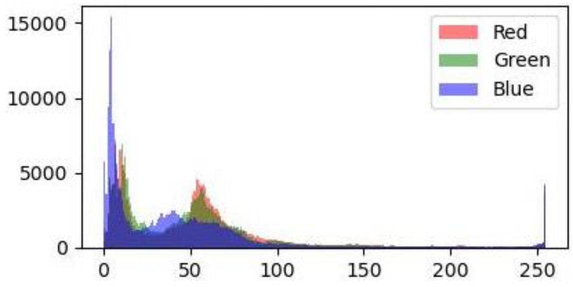

(d) HE RGB

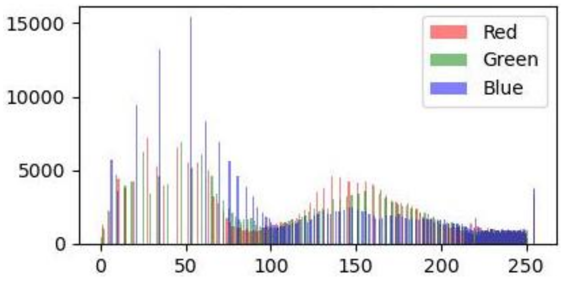
**Figure 16: Image Results from Sample 8**


(a) Original Gray


(b) HE Gray



(c) Original RGB



(d) HE RGB

**Figure 17: Histogram Results from Sample 8**

### 2.3 Discussion

As seen in the implementation above, the histogram equalization method is a simple and fast method of increasing the contrast of an image. Unlike other contrast enhancement methods, like gamma correction, HE requires no prior pre-processing or information about an image.

Because of the improved contrast, the details within an image have also been amplified. For example, in all the images, the structures are more prominent (i.e., edges and lines) after HE implementation. This helps improve to improve the visibility of an image for user analysis.

However, since HE is a global adjustment approach, it also considerably amplifies the noise within an image. This results in obvious artifacts seen in the equalized images above, which were otherwise not visible. The global enhancement approach also results in the loss of some finer details. This is illustrated in Figure 14 (c, d) where the structure of the lamp was lost in the equalized image.

Besides, HE is a purely mathematical approach. It does not afford any control for the user to fine-tune the corrective process. If the method produces an unnatural, noisy equalized images, then users must find another approach to get a more visually appealing image.

## 3. Improvement to HE: CLAHE

To circumvent some of these issues, we instead use the Contrast-Limited Adaptive Histogram Equalization (CLAHE) method.

### 3.1 Explanation on Implementation of HE Algorithm

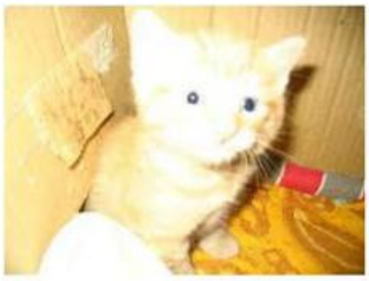
CLAHE has two parts to it. The first part is an implementation of adaptive histogram equalization (AHE). In AHE, a raw image is split into $N$ equal, non-overlapping regions called *tiles*. Then, HE will be done independently at each of the tile, with different histograms and CDF. Once every tile has been equalized, they are stitched together, typically using bilinear interpolation, to smooth the edges of the tiles as they come together.

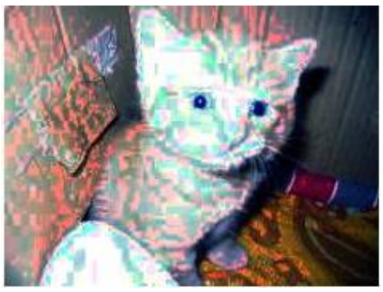
The second part is an improvement to AHE, where the histogram of each tile is being clipped. If a portion of the histogram exceeds the clip limit, it will clipped and redistributed uniformly across all the bins of the histogram. This limits the contrast that each tile can take.

For the implementation of CLAHE, we will use the *OpenCV* library. The ground-up implementation of CLAHE is beyond the scope of this report.

## 3.2 Image and Histogram Results from CLAHE

We will compare some of the RGB images obtained from CLAHE as opposed to HE.


(a) Original RGB


(b) HE RGB

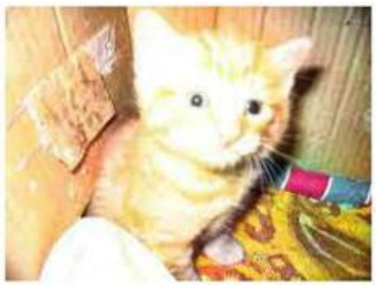
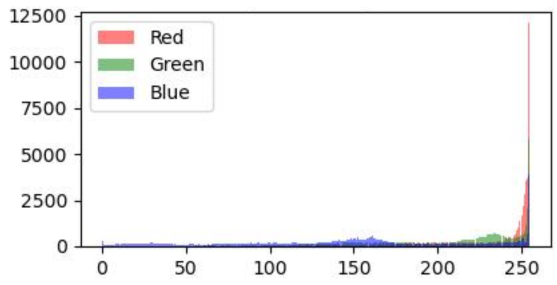
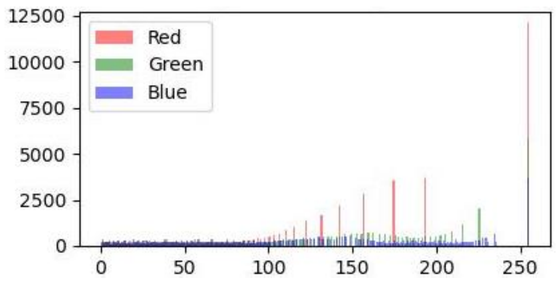

(c) CLAHE RGB

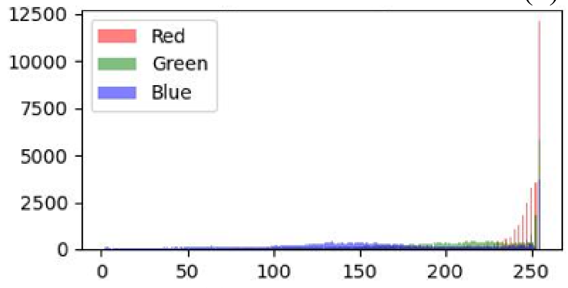
**Figure 18: RGB Images Results from Sample 2**



(a) Original RGB

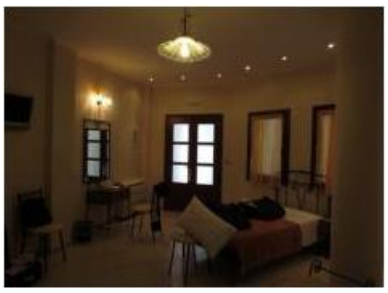


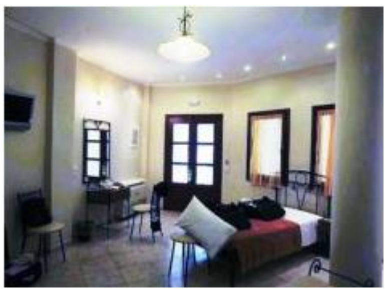
(b) HE RGB



(c) CLAHE RGB

**Figure 19: Histogram Results from Sample 2**


(a) Original RGB


(b) HE RGB

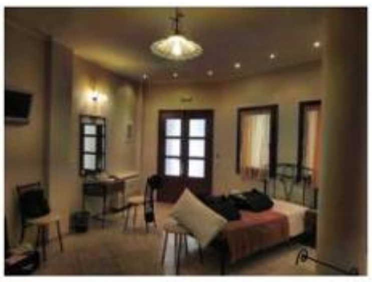
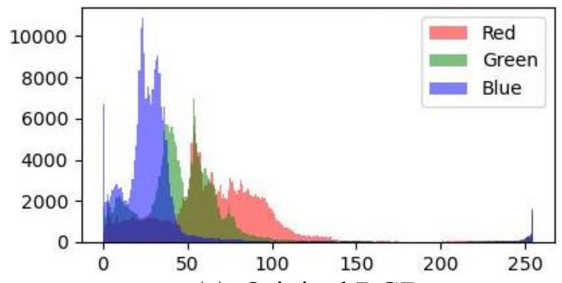
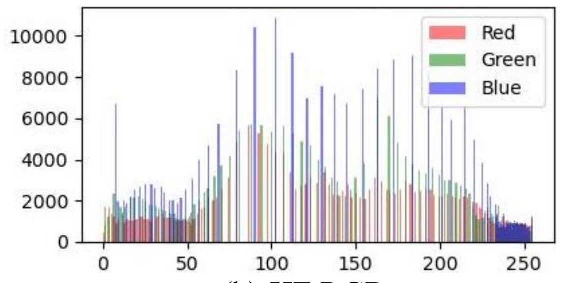

(c) CLAHE RGB

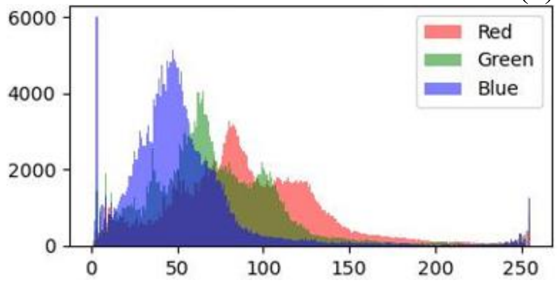
**Figure 20: RGB Images Results from Sample 7**



(a) Original RGB



(b) HE RGB



(c) CLAHE RGB

**Figure 21: Histogram Results from Sample 7**

## 3.3 Discussion

As seen above, the images produced by CLAHE has fewer artefacts compared to the images produced by HE. This suggests that the noise in CLAHE images is much less amplified, due to the contrast limiting effect of CLAHE, as well as applying HE locally rather than globally. In particular, the structure of the lamp in sample image 7 (Figure 20) is retained in CLAHE, while it was lost when HE was applied.

Interestingly, the histograms of the CLAHE images look like that of the original images, with corresponding regions of clustering. On the other hand, the histograms of the HE images look quite different from that of the original images, with many of the clustering being dispersed (see Figure 21). This difference is most likely a consequence of the local application of HE in CLAHE (i.e., the AHE part of CLAHE). This local contrast correction allows for different degree of HE to be applied to different segments of the images. Therefore, when put together, the general structure in the image is retained, as opposed to global HE, resulting in a similar distribution of histograms compared to the original images.

CLAHE seems to be able to address the drawbacks of HE. However, it is a more computationally expensive algorithm as compared to HE, as outlined in the steps of CLAHE.

## 4. Conclusion

There are several methods in the field of image processing to enhance the contrast of an image. One such methods is histogram equalization. This method is relatively straightforward and computationally efficient, as it redistributes the pixel intensities based on the global histogram. However, we have also seen some drawbacks of using this method, which includes distortion of image as well as amplified noise. We could improve on HE by considering the histograms of a local region separately, while also clipping the pixel intensity to reduce noise amplification. This method is called CLAHE, which proves to overcome some of the challenges of implementing HE, however it is a more computationally expensive one.