186

Activity 4: Enhancement of Color Images

Mary Chris Go / 2014 11122

Goal

to be able to restore faded photographs using various white balancing algorithms

Mary Chris Go / 2014 11122

Contrast Stretching

Also known as ‘normalization’. It improves the image by ‘stretching’ the range of intensity values.

Gray World Algorithm

A white balanced method that sees your image as a neutral gray. It produces an estimation of illumination by averaging each of the RGB channel.

White Patch Algorithm

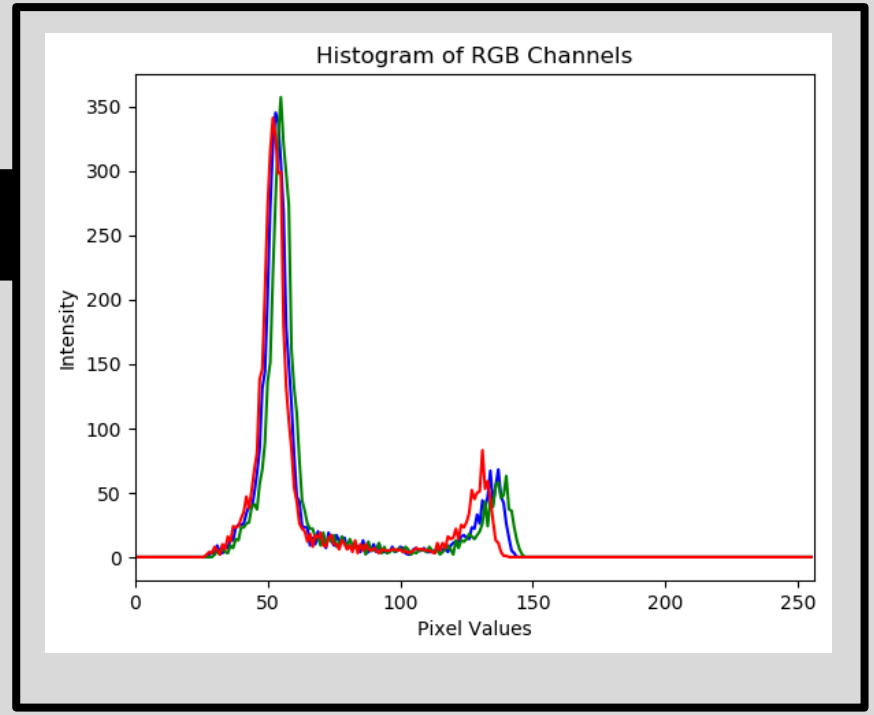
This algorithm assumes the perfect reflectance shows the maximum response of an image.

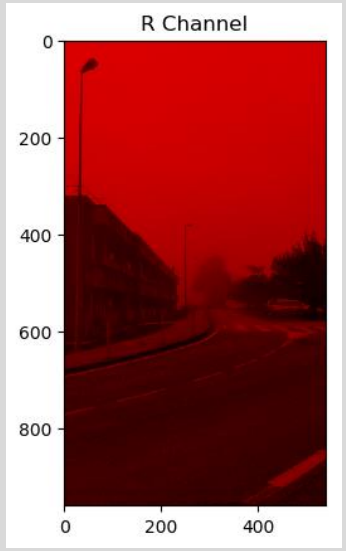
Mary Chris Go / 2014 11122

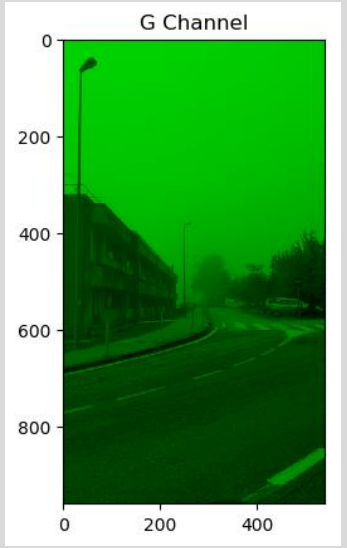
Original Image

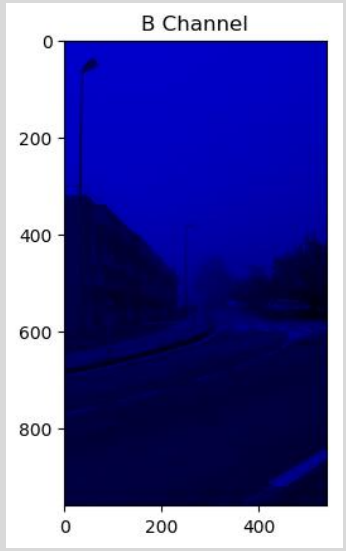
The chosen picture was from Vigo, Spain. A faded feature was shown because of the weather (Grabbed from a good friend).

Minimum: 5
Maximum: 225





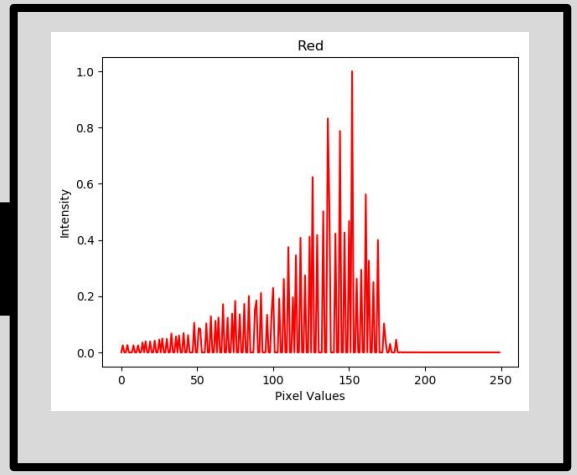


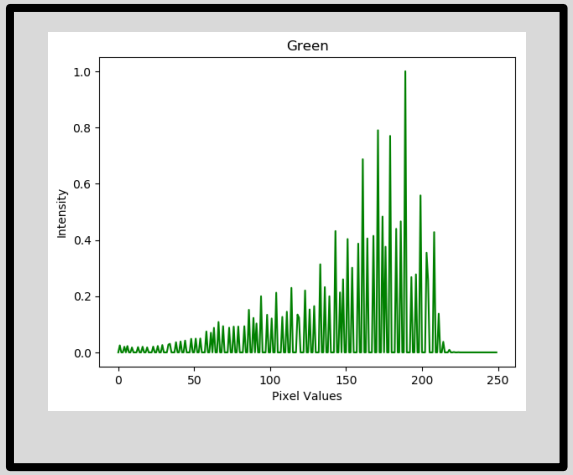


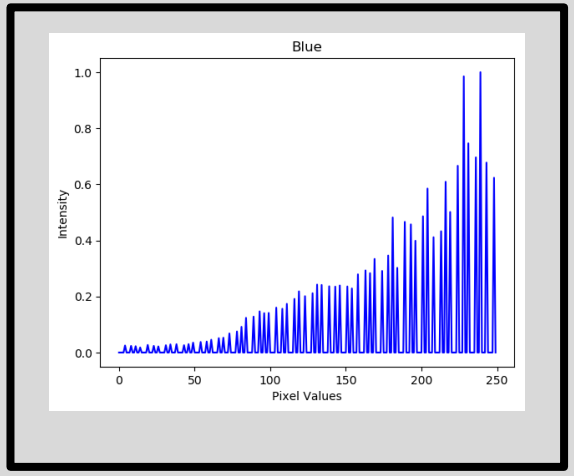
I got the histogram per channel and superimposed them in one graph. As we can observe, the peaks of the histogram were located near the black part which is the left area. We can see this from the original image where half of the picture was dark.

Also, individual RGB channels were also shown

RGB Normalized Histograms



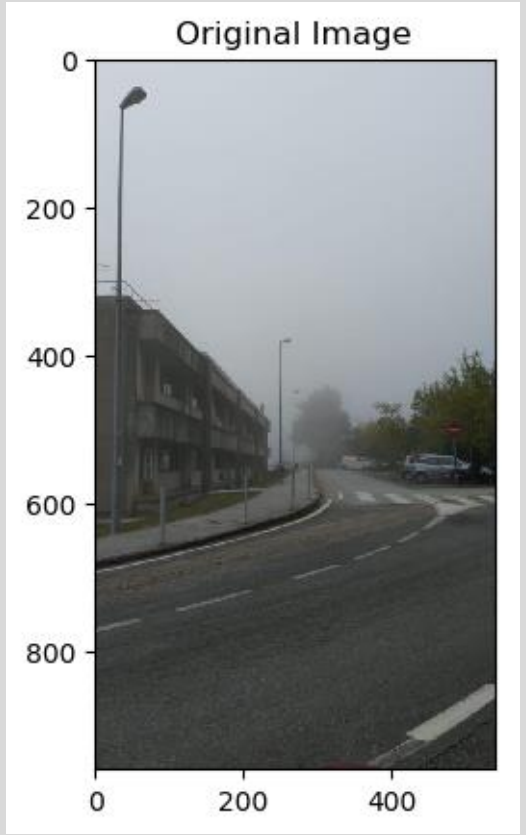


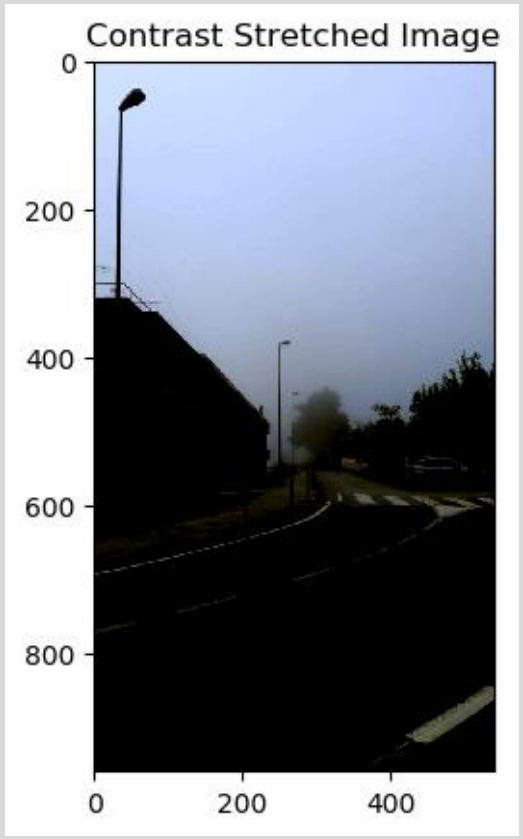



```
def normalizeRed(intensity):
    iI      = intensity
    minI    = 86
    maxI    = 230
    min0    = 0
    max0    = 255
    i0      = (iI-minI)*(((max0-min0)/(maxI-minI))+min0)
    return i0
```

The following code was used for normalizing the red channel. The same method was applied for green and blue channels. These normalized histograms were merge using **Image.merge.** It also performed contrast stretching per histogram.

Mary Chris Go / 2014 11122





After merging the channels, this was the result for contrast stretched image.


```python
def Gray_Algo (G):
    G_gray = np.copy(G)
    R1,G1,B1 = cv2.split(G_gray)
    H1 = G_gray.shape[0]
    W1 = G_gray.shape[1]
    spec_R = np.sum(R1)/(H1*W1)
    spec_G = np.sum(G1)/(H1*W1)
    spec_B = np.sum(B1)/(H1*W1)
    scale = (spec_R+spec_G+spec_B)/3
    R_ave = (np.sum(R1)/(R1.shape[0]*R1.shape[1]))
    G_ave = (np.sum(G1)/(G1.shape[0]*G1.shape[1]))
    B_ave = (np.sum(B1)/(B1.shape[0]*B1.shape[1]))
    R_wb = (R1/R_ave)*scale
    G_wb = (G1/G_ave)*scale
    B_wb = (B1/B_ave)*scale
    G_gray[ : , : , 0] = R_wb
    G_gray[ : , : , 1] = G_wb
    G_gray[ : , : , 2] = B_wb
    return G_gray
```

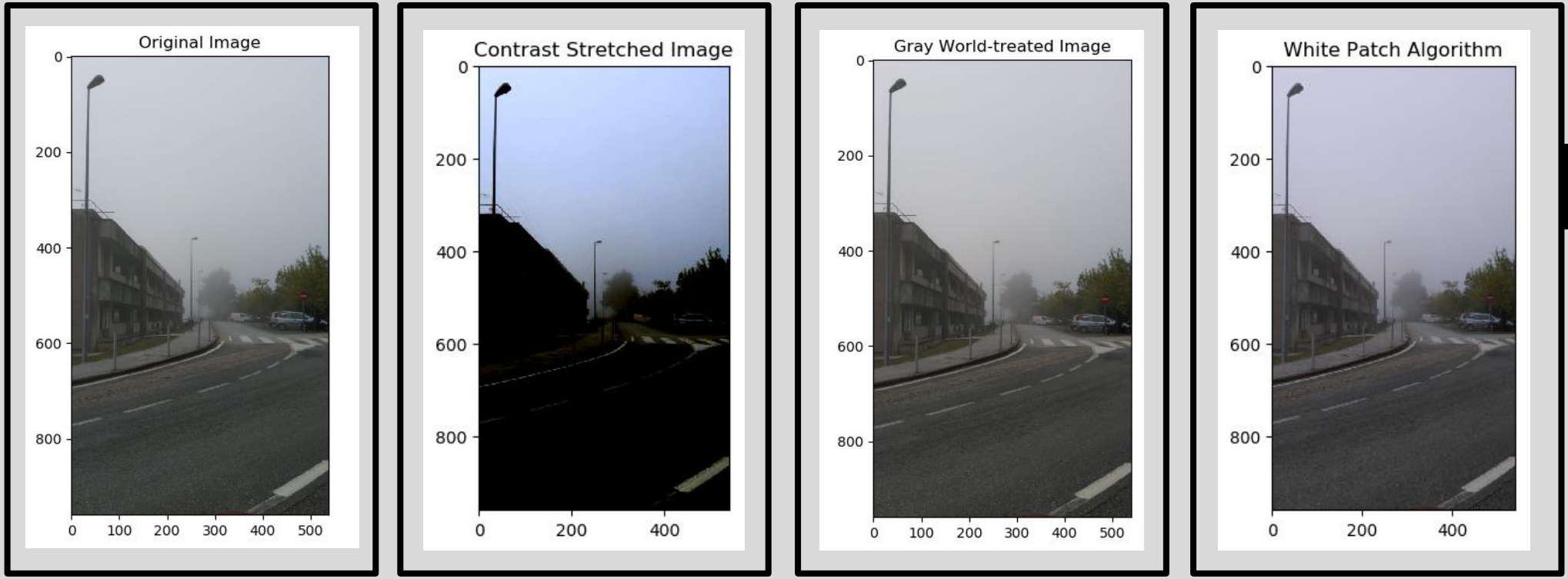
For the Gray World Algorithm, the directions from the were implemented using Python. Each channel was divided by their respective averages. A snippet code was attached.

```python
#original image
G_gray = cv2.cvtColor(cv2.imread('2.jpg'), cv2.COLOR_BGR2RGB)
R,G,B =cv2.split(G_gray)

#white image
M = cv2.cvtColor(cv2.imread('white.jpg'), cv2.COLOR_BGR2RGB)
R1,G1,B1 = cv2.split(M)

F = np.copy(G_gray)
H1 = M.shape[0]
W1 = M.shape[1]
spec_R = np.sum(R1)/(H1*W1)
spec_G = np.sum(G1)/(H1*W1)
spec_B = np.sum(B1)/(H1*W1)
scale = (spec_R+spec_G+spec_B)/3

R_ave = (np.sum(R1)/(R1.shape[0]*R1.shape[1]))
G_ave = (np.sum(G1)/(G1.shape[0]*G1.shape[1]))
B_ave = (np.sum(B1)/(B1.shape[0]*B1.shape[1]))

R_wb = (R/R_ave)*scale
G_wb = (G/G_ave)*scale
B_wb = (B/B_ave)*scale
F[ : , : , 0] = R_wb
F[ : , : , 1] = G_wb
F[ : , : , 2] = B_wb
```

For the white patch algorithm, the chosen white region was encircled. Average RGB channels of these images were obtained. Each channel of the original image were divided with the average RGB channels.

Mary Chris Go / 2014 11122

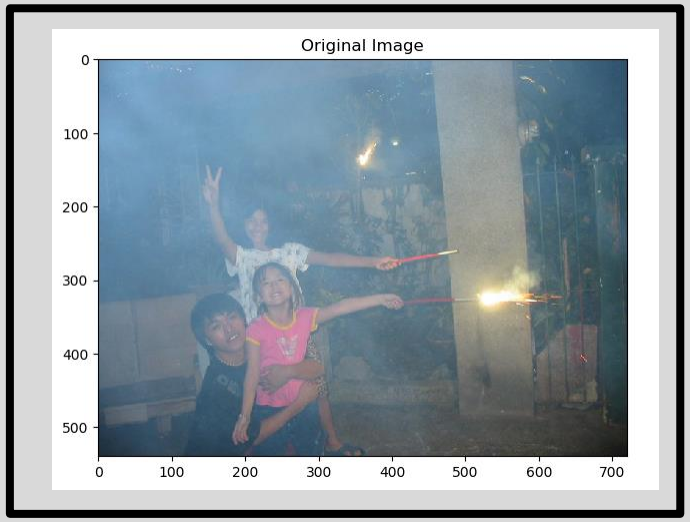
Results

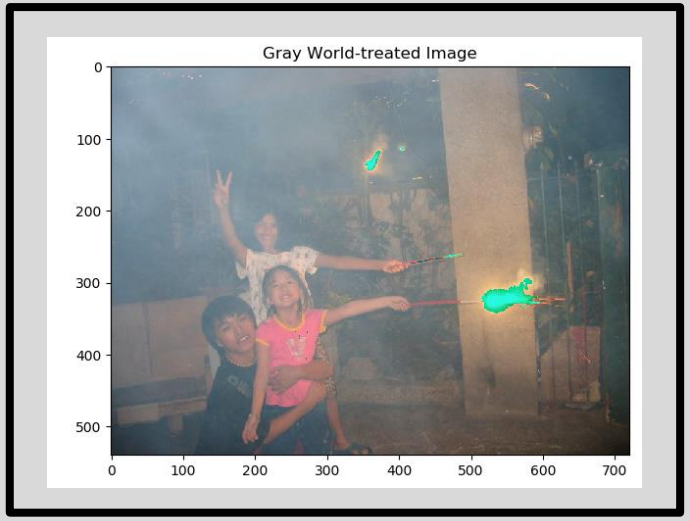


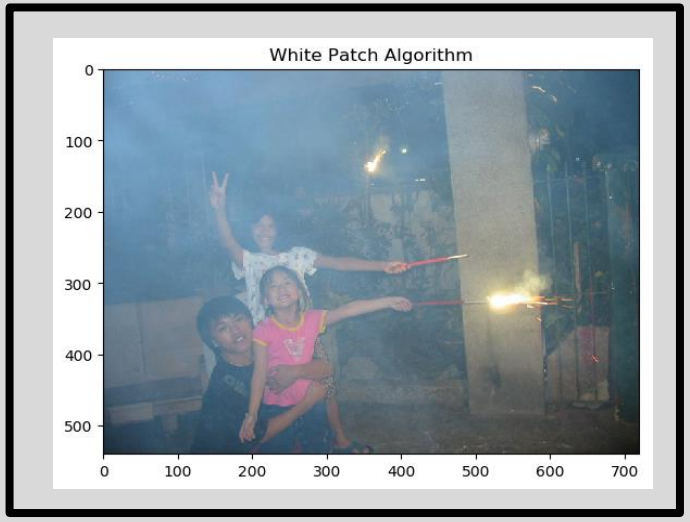
The contrast stretched image intensified the colors too much. The ‘gray sky’ was shown to be blue. The gray world algorithm treated my original picture well. It did intensify the image but not so much unlike the contrast stretching. You can still identify different objects here but it turned a bit yellow. Lastly, the white patch algorithm was the best treated image. It was able to balance the image and develop its colors.

Mary Chris Go / 2014 11122

Other Input Image







In this image, the contrast stretched algorithm was the best in treating this kind of image.

Mary Chris Go / 2014 11122

Summary

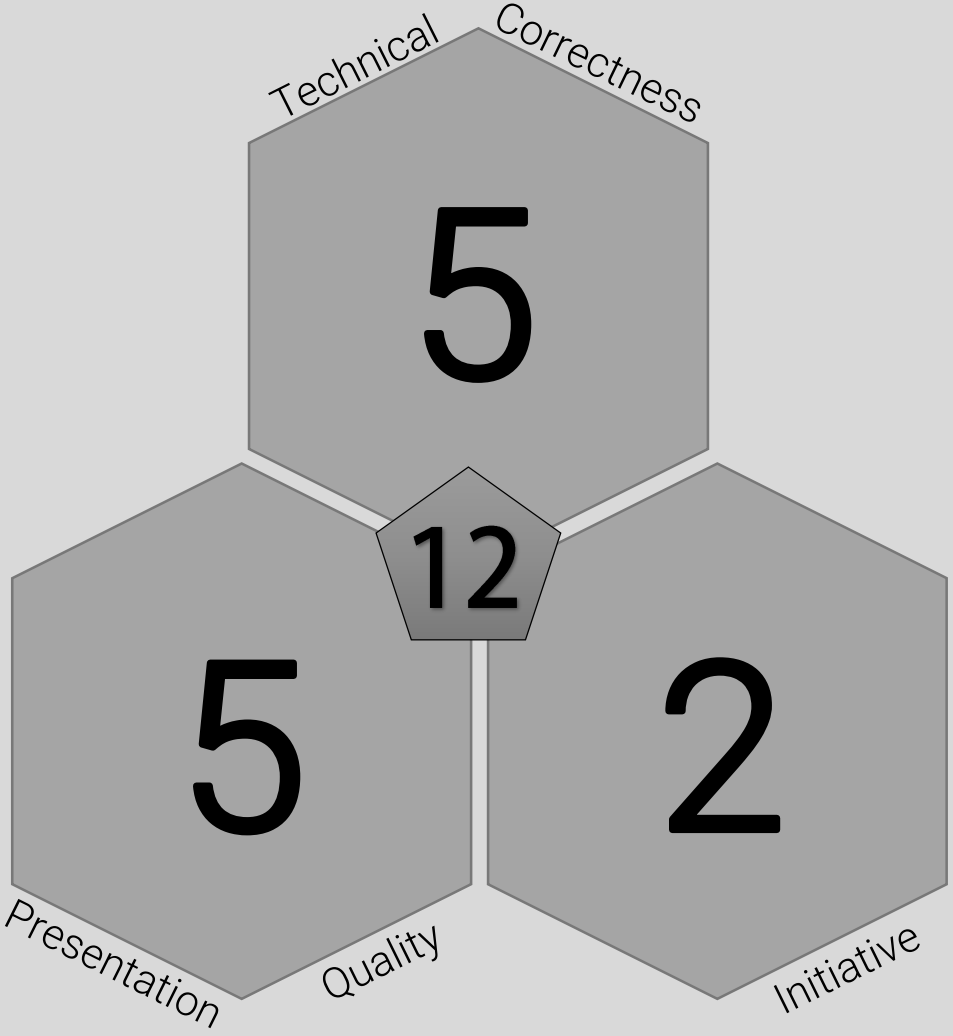
All in all, the white patch algorithm was the best in treating my first original image. Also, I tried to compare my results with other input image and I observed that it shows different results depending on your original image. For the second one, the contrast stretching was the best.

The concept of “filters” was emerging in this world of social media prevalence. From this activity, I’ve learned when you like to filter your image, the original image must be taken into account. There is no such thing as ‘best filter’. It all depends on the input image.

This activity was very applicable in today’s generation. It was fun doing it. Hoping to have more activities about it.

Mary Chris Go / 2014 11122

Self-Evaluation



References

- (n.d.). Illuminant Estimation: Gray World. Retrieved from https://web.stanford.edu/~sujason/ColorBalancing/grayworld.html
- (n.d.). Contrast Stretching. Retrieved from https://homepages.inf.ed.ac.uk/rbf/HIPR2/stretch.htm
- Rizzi, A., Gatta, C., & Marini, D. (n.d.). Color correction between Gray World and White Patch. Retrieved from https://www.researchgate.net/publication/228810816_Color_correction_between_Gray_World_and_White_Patch
- Soriano, M. Enhancement of Color Images, Applied Physics 186