TDS 3651 VISUAL INFORMATION PROCESSING

ASSIGNMENT 1 REPORT

LECTURER: DR WONG LAI KUAN

Prepared by

Name	ID	Email
Balaganabathy A/L Krishnanmoorthy	1161304121	gana.ganabathy@gmail.com

Abstract

Satellite image enhancement is the technique which is most widely used in the field of satellite image processing to enhance the visualization of the features. Satellite images are captured from a very far distance, thus they contain too much noise and distortion due to atmospheric barriers. It is crucial to enhance the restored image before using it. In this assignment, a series of methods for satellite image enhancement via histogram equalization, gamma correction and saturation adjustment are discussed in detail, as well as experimental results of using different histogram equalization techniques and different parameters for all three methods are discussed.

Introduction

Image enhancement techniques are divided into three broad categories:

1. Spatial Domain Method
2. Frequency-Domain Methods
3. Color enhancement

Various research papers are mostly focused only on the first two methods of enhancement. However, sometimes we need to convert gray level image into the RGB color image and eventually, we have to apply the color enhancement of the images too. In the spatial domain, the image is filtered via pixel to pixel whereas in the frequency domain, images are converted into a signal and later discrete transforms are applied i.e. Fourier, Laplacian or wavelet transforms.

In this assignment, we mainly implement spatial domain methods which mainly focus on point to point manipulation of every single pixel. Histogram equalization (HE) flattens and stretches the dynamic range of the image's histogram and results in overall contrast improvement. In this assignment, we apply **Contrast Limited Adaptive Histogram Equalization (CLAHE)** which is a variant of adaptive histogram equalization in which the contrast amplification is limited, so as to minimize noise amplification.

In **gamma correction**, pixel values are mapped to the range between 0 and 1 are raised to the power of chosen positive exponent. The exponent in gamma correction is chosen for each image such that the overall accuracy of image similarity between output and ground truth image is highest. Finally, the **image saturation is enhanced** by a saturation factor to make the image more vibrant. The saturation factor is chosen such that the overall accuracy of image similarity between output and ground truth image is highest. After this step, the image is fully enhanced. The next section provides in depth details of the methodologies used.

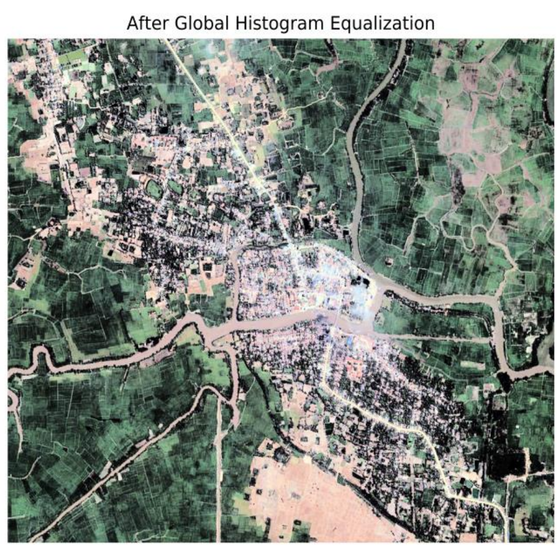
<u>Description of method</u>

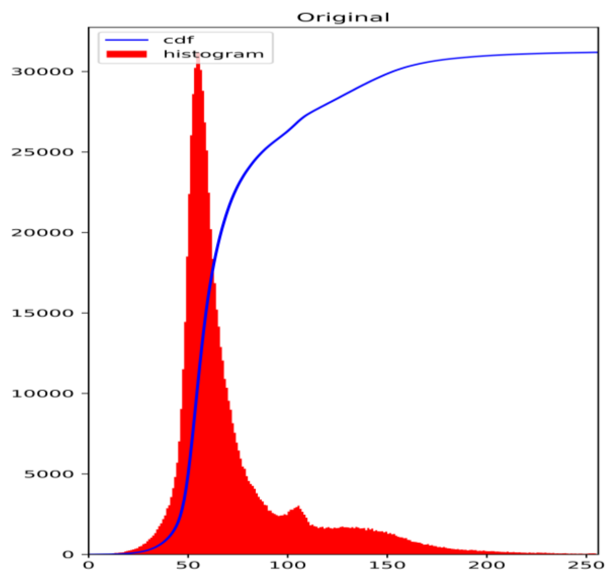
1) CLAHE

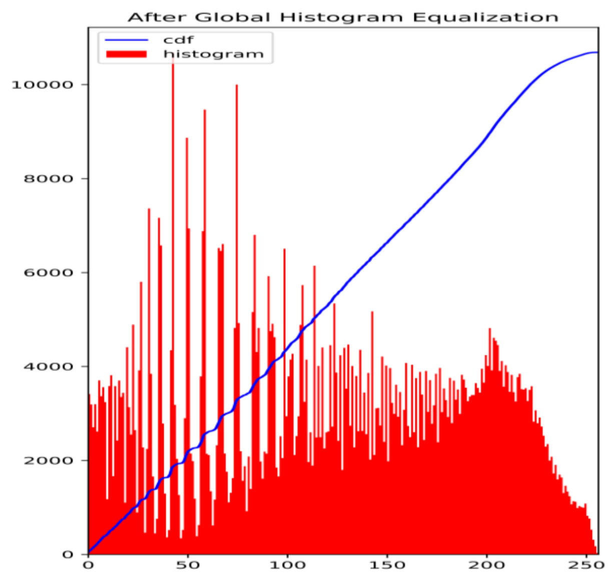
A color histogram of an image represents the number of pixels in each type of color channel. Histogram equalization cannot be applied separately to the Red, Green and Blue components of the image as it results in dramatic changes in the image's color balance. Therefore, we first convert the input image to **HSV color space**, and then histogram equalization is applied to the **value** channel without resulting in changes to the **hue** and **saturation** of the image.

Histogram equalization is a technique used for contrast adjustment using the image's histogram. It is not necessary that contrast will always be enhanced in this technique. There may be some cases were histogram equalization can be worse. In those cases, the contrast is reduced. Histogram equalization is efficient whenever image histogram is confined to a particular region. It will not work well in places where there are large intensity variations where histogram covers a large region, i.e. both bright and dark pixels are present.

The aforementioned histogram equalization considers the global contrast of the image. In many cases, it is not a good idea as mentioned. For instance, the figure below shows an input image (**input17 from testset 1**), its transformation after global histogram equalization and their respective histogram plots.

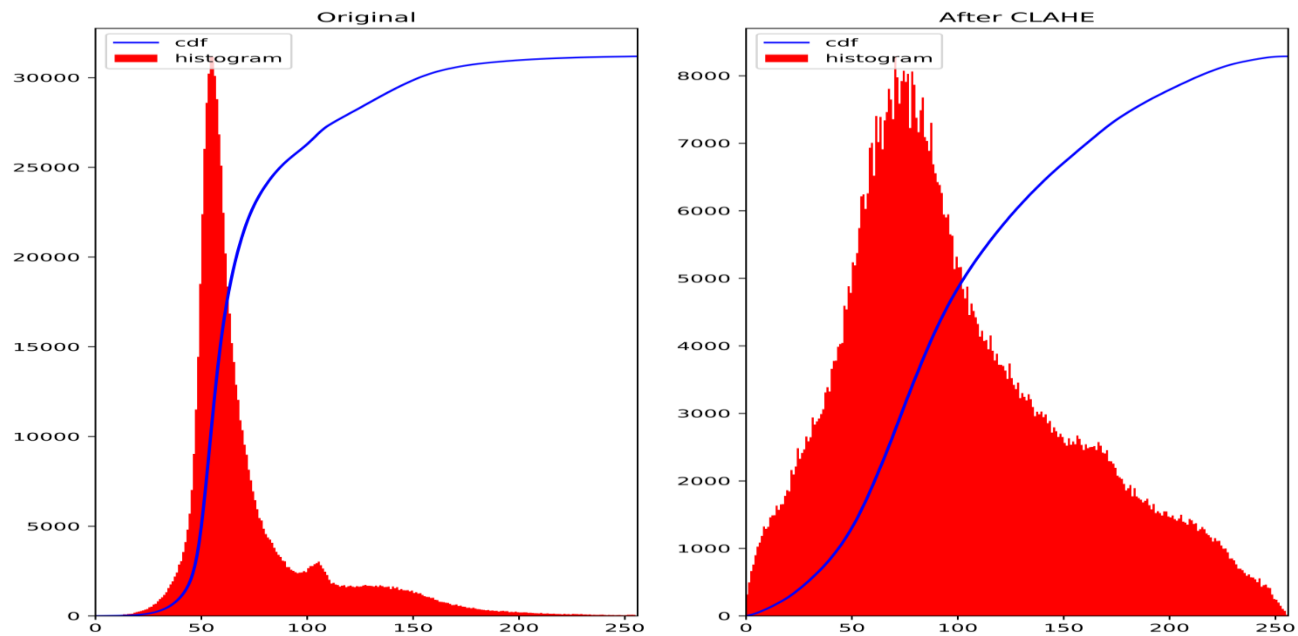






It is true that the background contrast has improved after histogram equalization. However, compare details in both images. We lost most of the information in original image due to over-brightness. It is because the resulting histogram is not confined to a particular region.

So to solve this problem, **adaptive histogram equalization** can be used. In this, input image is divided into small blocks called "tiles" (tileSize is 8x8 by default in **OpenCV**). Then each of these blocks are histogram equalized as usual and used to redistribute the lightness values of image. Hence, it is suitable for enhancing the local contrast and definition of edges in each region of an image. So in a small area, histogram would confine to a small region (unless there is noise). If noise is there, it will be amplified. To avoid this, we instead apply **contrast limiting (CLAHE)** in our assignment. If any histogram bin is above the specified **contrast limit** (by default 40 in **OpenCV**), those pixels are clipped and distributed uniformly to other bins before applying histogram equalization.



The value at which the histogram is clipped, the so-called clip limit, depends on the normalization of the histogram and thereby on the size of the neighbourhood region. Common values limit the resulting amplification to between 3 and 4. . The clipLimit used for this method is **4.0** which is commonly used and works very well for both test sets. After equalization, to remove artifacts in tile borders, bilinear interpolation is applied automatically. Through this, we have prevented over amplification of noise and retain the details in the original image. If we compare the result of **CLAHE** with **global histogram equalization**, the difference is obvious.

2) Power Law Transformations (Gamma Correction)

When twice the amount of photons hit the sensor of a digital camera, it obtains twice the signal (a linear relationship). However, that's not how our human eyes function. Instead, we perceive double the amount of light as **only** a fraction brighter (a non-linear relationship). Furthermore, our eyes are also much more sensitive to changes in dark tones than brighter tones (another non-linear relationship).

In order to resolve this, we can apply gamma correction, a translation between the sensitivity of our eyes and sensors of a camera. If images are not gamma-encoded, too many bits assigned for the bright tones that humans cannot differentiate and too few bits for the dark tones. Through gamma encoding, we remove this artifact. Images which are not properly corrected can look either bleached out, or too dark.

Gamma correction is also known as the **Power Law Transform**. First, our image pixel intensities must be scaled from the range [0, 255] to [0, 1.0]. From there, we obtain our gamma corrected image by applying the following equation:

O = I ^ (1 / G)

Where **I** is our input image and **G** is our gamma value. The output image **O** is then scaled back to the range [0, 255].

Gamma values **< 1** will shift the image towards the darker end of the spectrum while **values > 1** will make the image appear brighter. A gamma value of **G=1** will have no effect on the input image. We applied gamma correction using built in functions in **OpenCV**. All we need to do is build a table (i.e. dictionary) that maps the input pixel values to the output gamma corrected values. **OpenCV** can then take this table and quickly determine the output value for a given pixel in O (1) time. First, we build this lookup table by looping over all pixel values in the range [0, 255]. The pixel value is then scaled to the range [0, 1.0] followed by being raised to the power of the inverse gamma. This value is then stored in the table.

Lastly, we applied the **cv2.LUT( )** function which takes the input image and the table to find the correct mappings for each pixel operation. In our case, we need to reduce the brightness of image slightly due to result of histogram equalization. Hence, the range of gamma values tested is **[0.20, 0.80]**. Through experimentation, we found out that gamma = **0.83** yield the highest similarity score between output image and ground truth. Hence, this value is fixed for **adjust_gamma ( )** method in **imageEnhance.py**. The result below shows input image before and after gamma correction:

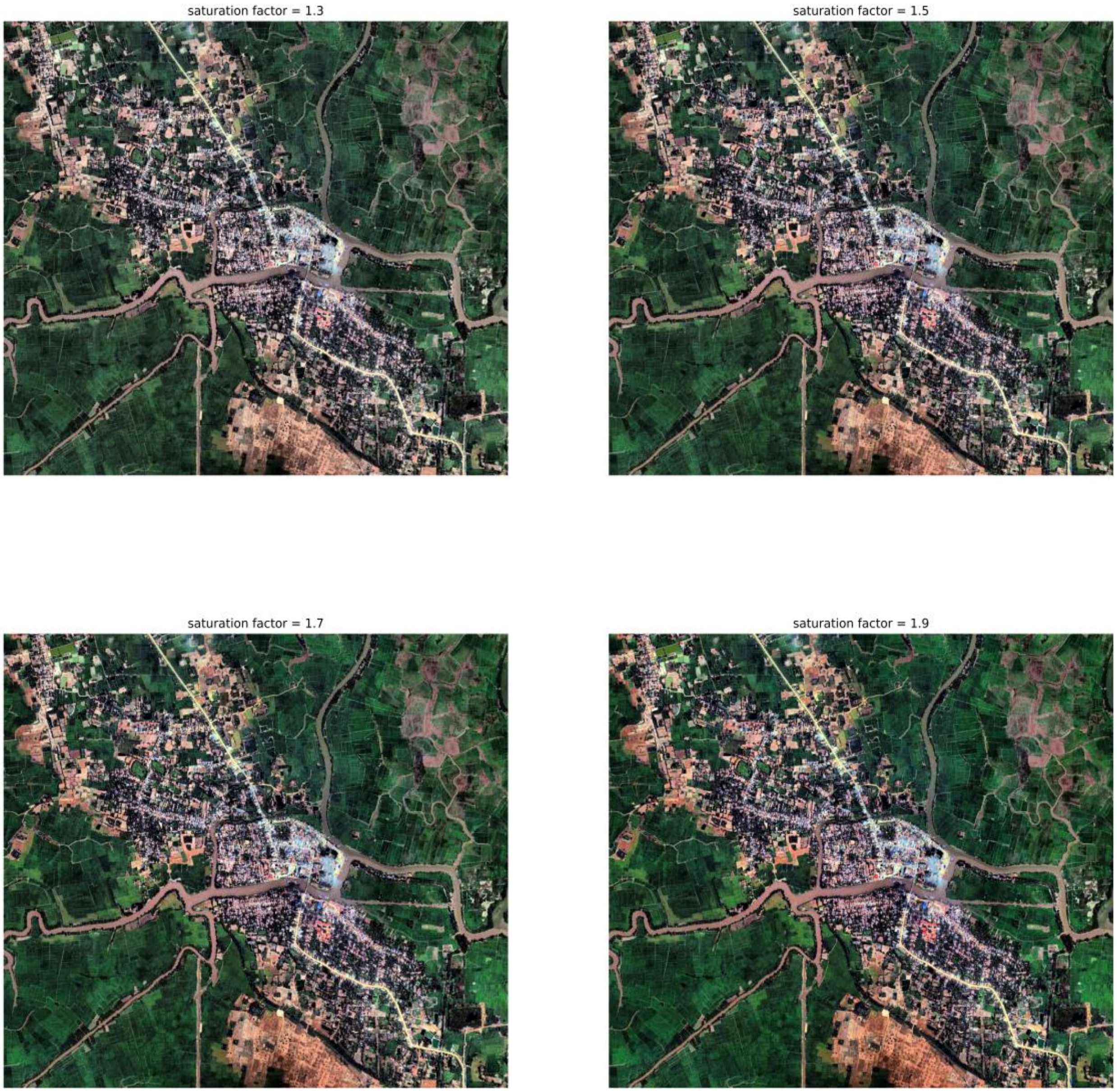
gamma = 0.2

gamma = 0.4

3) Saturation adjustment

Saturation refers to the intensity of a color. The higher the saturation of a color, the more vivid it is. The lower the saturation of a color, the closer it is to gray. Lowering the saturation of a photo can have a "**muting**" or **calming effect**, while increasing it can increase the feel of the **vividness** of the scene. It is important not to over-saturate a photo, as sometimes it creates unnatural color spill-over. As our last step, we increased the saturation of our image to make it more vibrant and colorful. We achieved this by using built in method called **adjust_saturation( )** in **tensorflow image module**. This is a convenient technique that converts RGB images to float representation, converts them to HSV, add an offset to the saturation channel, converts back to RGB and then back to the original data type.

This function requires two parameters: **image** and **saturation_factor**. Image is an RGB or other format images. Image saturation is adjusted by converting the image to HSV and multiplying the saturation (S) channel by **saturation_factor**. The images are then converted back to original format. Since we only want to enhance the saturation slightly, the range of saturation_factor tested is **[1.30, 1.90]**. Through experimentation we found out that **saturation_factor = 1.65** and **saturation_factor** = **1.35** yield the highest similarity score between output image and ground truth for **testset1** and **testset2** respectively. Hence, this value has to be changed accordingly in **adjust_saturation ( )** method inside **imageEnhance.py.** The result below shows input image before and after saturation adjustment for saturation factor = **1.3, 1.5, 1.7, 1.9**.


saturation factor = 1.3
saturation factor = 1.5
saturation factor = 1.7
saturation factor = 1.9

<u>Results & Analysis</u>

In this section we will provide visualizations to show before and after transformation for each successive method for both test set 1 and 2. CLAHE and gamma correction done together before applying saturation adjustment. Since there are many images for each set we only provide visualization for a subset of images. (**15 images** for testset 1 and **8 images** for testset 2). For each testset, table of results showing similarity (calculated from **evaluate.py**) are also provided to show how the similarity measure increases after each method is applied.

Test set 1

After CLAHE and Gamma correction

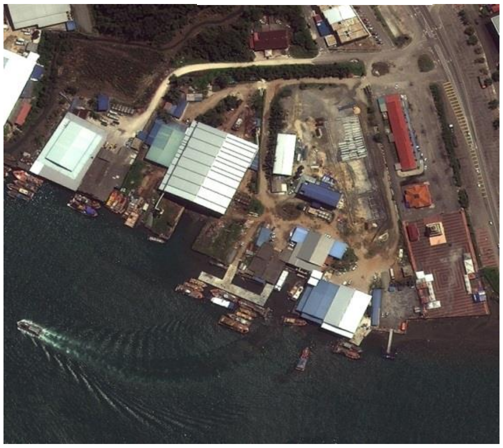
Image	Similarity Score
1	0.7168
2	0.6655
3	0.7746
4	0.7553
5	0.7863
6	0.7196
7	0.7186
8	0.6595
9	0.8348
10	0.7102
11	0.7147
12	0.8243
13	0.8272
14	0.7704
15	0.7405
16	0.8312
17	0.7098
18	0.845
19	0.8502
20	0.8126
21	0.8487
22	0.881
23	0.7307
24	0.8354
25	0.8193
26	0.8708
27	0.8226
28	0.8324
29	0.8182
30	0.8178
All	0.7848

After saturation adjustment

Image	Similarity Score
1	0.7101
2	0.8067
3	0.8356
4	0.8783
5	0.8726
6	0.8125
7	0.8088
8	0.7357
9	0.9115
10	0.8046
11	0.8235
12	0.8683
13	0.8748
14	0.8678
15	0.6965
16	0.8042
17	0.8221
18	0.9065
19	0.7376
20	0.8145
21	0.8679
22	0.8288
23	0.798
24	0.9224
25	0.9127
26	0.8586
27	0.8586
28	0.8741
29	0.862
30	0.8296
All	0.8335

As we can see from table, the average similarity score **is 0.785** after applying CLAHE and gamma correction. The score then increased to **0.834** after applying saturation adjustment. Not all individual images' score has increased since some of them decreased. However, the number and amount of increase is higher compared to number and amount of decrease resulting in overall higher average. The part where similarity decreased will be discussed further in suggestion for improvement. The section below shows transformation visualization for selected images from testset1.

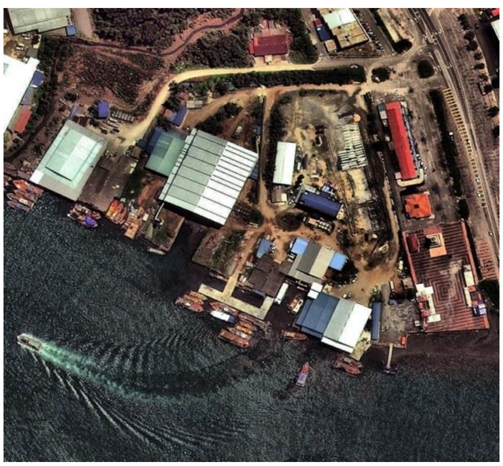
Input 1



Input 3

Input 5

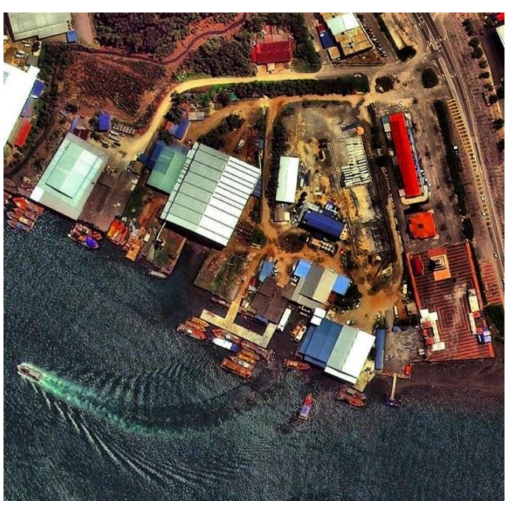
Output1 (i)



Output 3 (i)

Output 5 (i)

Output1 (ii)



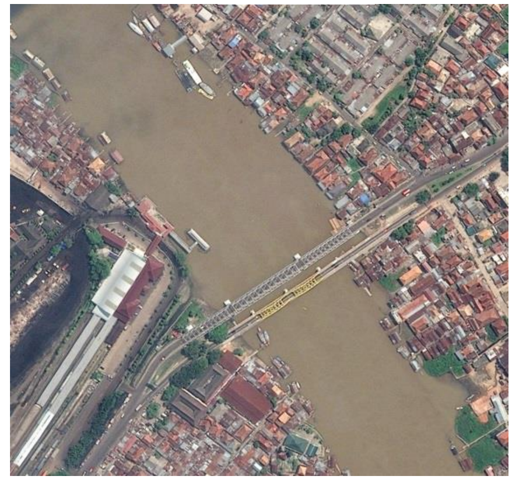
Output3 (ii)

Output5 (ii)

Input 7

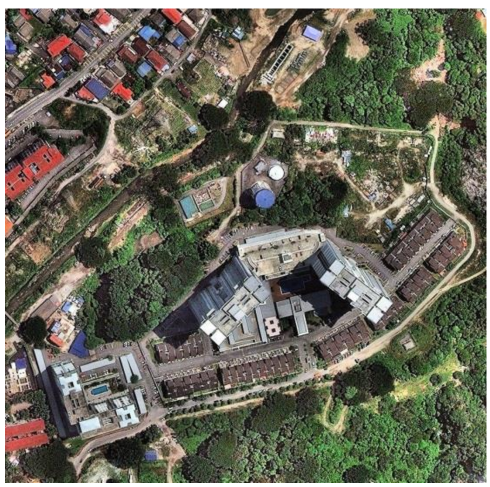
Input 9

Input 11



Output 7 (i)

Output 9 (i)

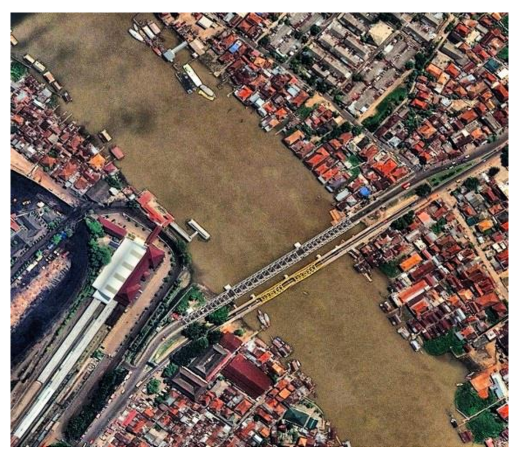


Output 11 (i)

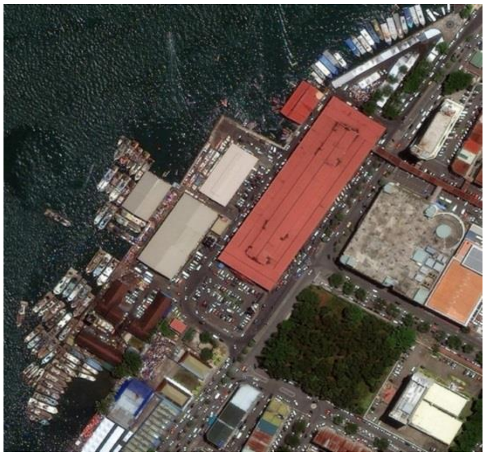
Output 7 (ii)

Output 9 (ii)

Output 11 (ii)

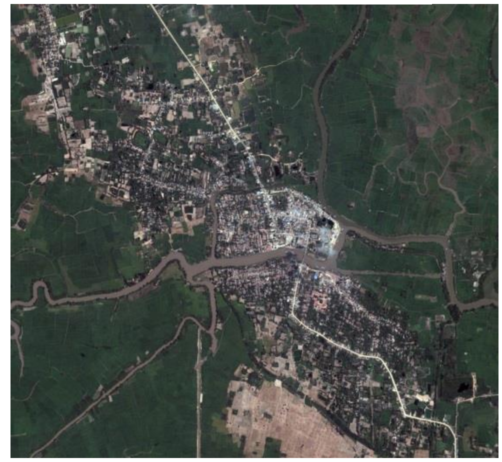


Input 13



Input 15

Input 17



Output 13 (i)

Output 15 (i)

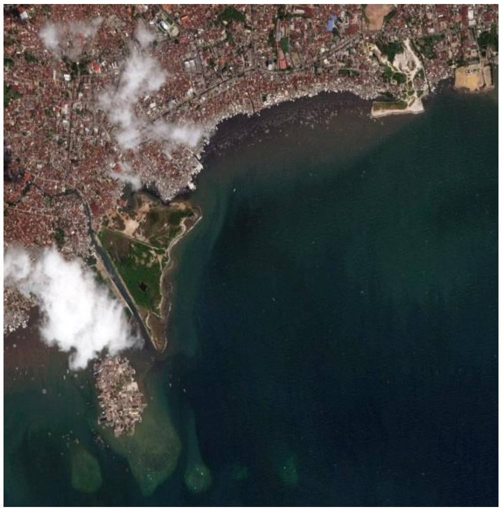
Output 17 (i)

Output 13 (ii)

Output 15 (ii)

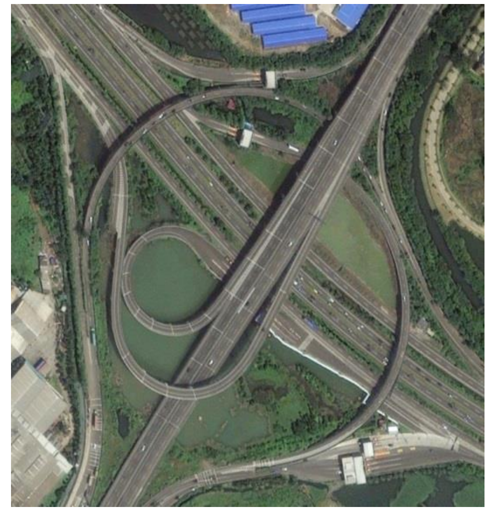
Output 17 (ii)

Input 19



Input 21

Input 23



Output 19 (i)

Output 21 (i)

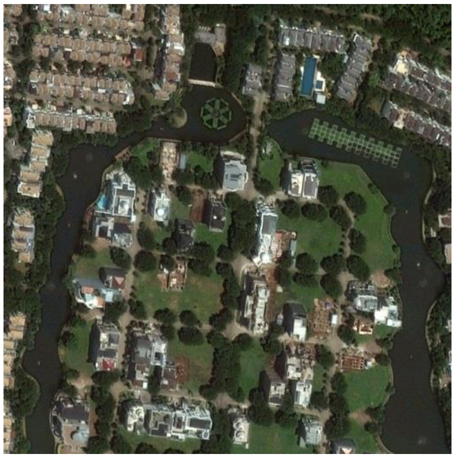
Output 23 (i)

Output 19 (ii)

Output 21 (ii)

Output 23 (ii)

Input 25



Input 27

Input 29

Output 25 (i)

Output 27 (i)

Output 29 (i)

Output 25 (ii)

Output 27 (ii)

Output 29 (ii)

Test set 2:

After CLAHE and gamma correction

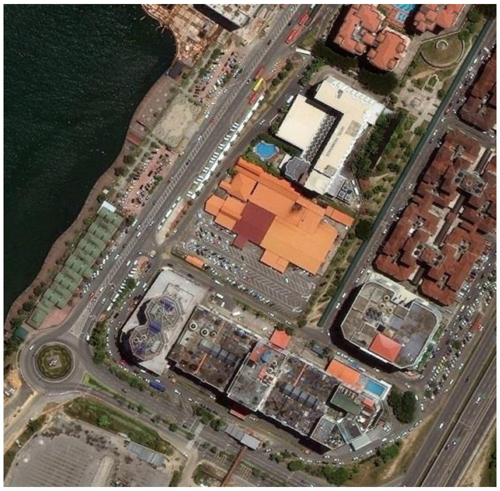
Image	Similarity Score
1	0.8605
2	0.8216
3	0.6391
4	0.8108
5	0.7453
6	0.7553
7	0.7423
8	0.7339
9	0.8428
10	0.7571
11	0.618
12	0.6096
13	0.6991
14	0.7847
15	0.8076
All	0.7485

After saturation adjustment

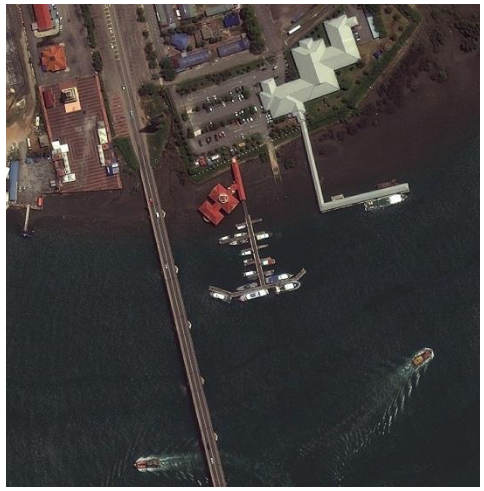
Image	Similarity Score
1	0.852
2	0.8184
3	0.6355
4	0.8507
5	0.8002
6	0.7195
7	0.8158
8	0.7748
9	0.8688
10	0.7928
11	0.6686
12	0.57
13	0.7436
14	0.8591
15	0.8064
All	0.7717

As we can see from table above, the average similarity score is **0.749** after applying CLAHE and gamma correction. The score then increased to **0.772** after applying saturation adjustment. The overall similarity is much lower compared to testset 1 since images in testset 2 has more variations and captured under challenging scenarios. Same as testset 1, not all individual images' score in testset 2 has increased since some of them decreased. However, the number and amount of increase is higher compared to number and amount of decrease resulting in overall higher average similarity score. The part where similarity decreased will be discussed further in suggestion for improvement. The section below shows transformation visualization for selected images from testset2.

Input 1



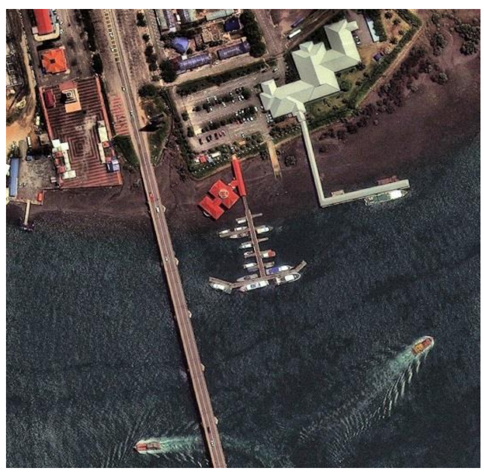
Input 3



Input 5

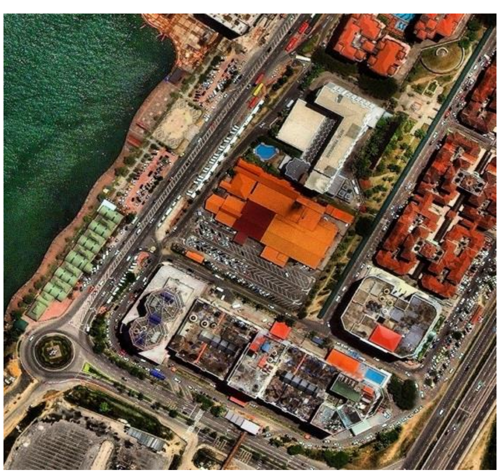
Output 1 (i)

Output 3(i)

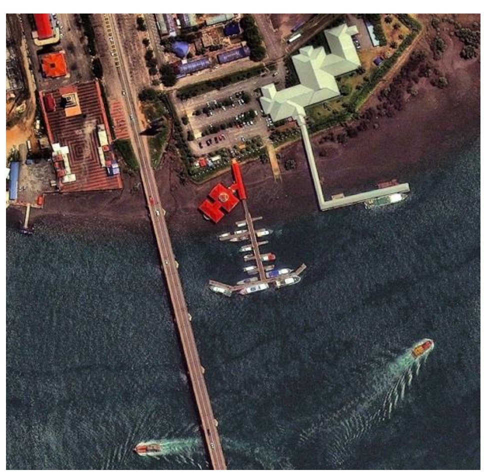


Output 5 (ii)

Output 1 (ii)



Output 3 (ii)

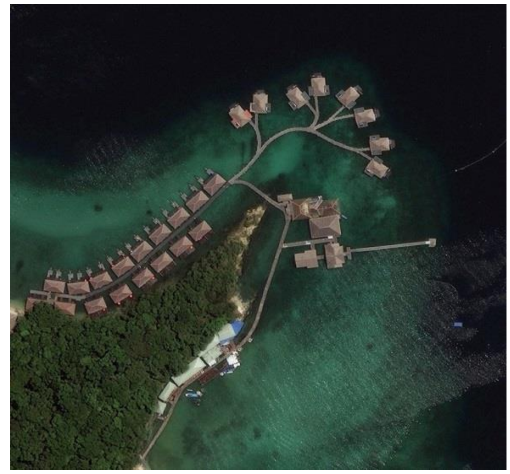


Output 5 (ii)

Input 7

Input 9

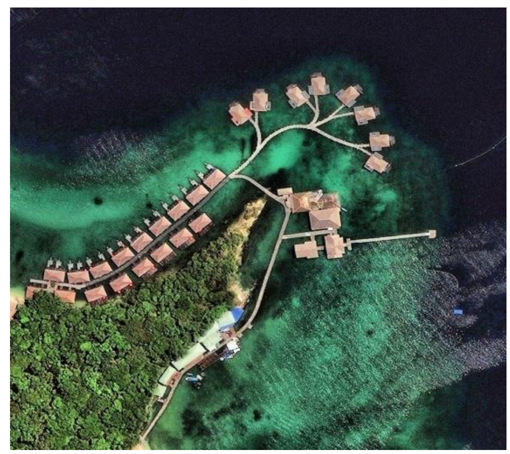
Input 11



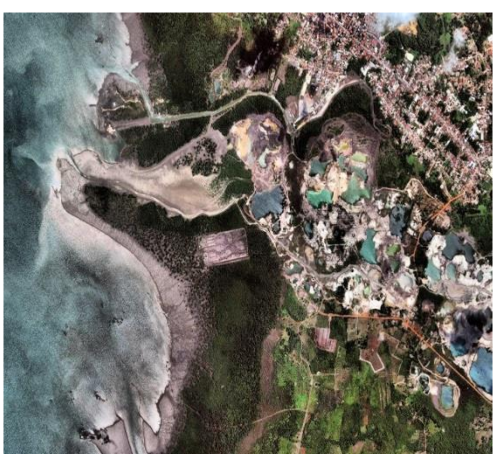
Output 7 (i)

Output 9 (i)

Output 11 (i)



Output 7 (ii)

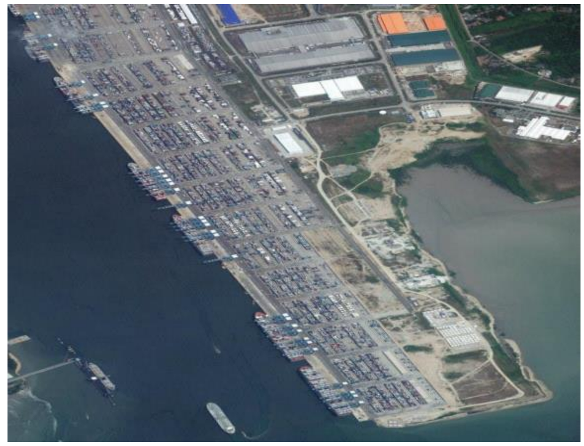


Output 9 (ii)

Output 11 (ii)

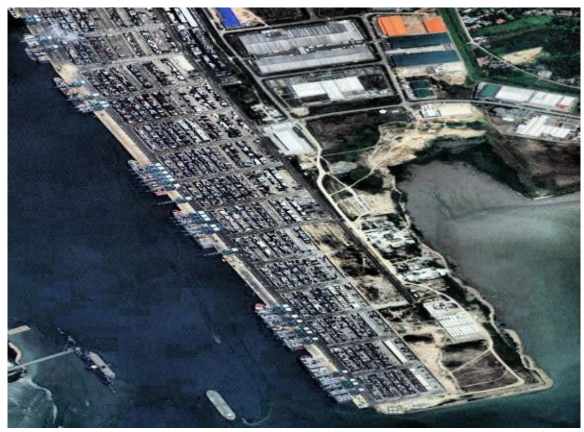
Input 13

Input 15



Output 13 (i)

Output 15 (i)



Output 13 (ii)

Output 15 (ii)

Suggestions for Improvement

There are plenty of rooms for improvement in this assignment. First of all, the gamma values used in gamma correction (**0.83**) is calculated through trial and error. We could do this because we had the ground truth image in hand. In real world scenario this is not possible. Hence, we need a more robust method that automatically determines the gamma value. In this case we could use **Efficient Contrast Enhancement Using Adaptive Gamma Correction with Weighting Distribution (AGCWD) [1]**. This paper proposes an efficient method to modify histograms and enhance contrast in digital images. They present an automatic transformation technique that improves the brightness of dimmed images via the gamma correction and probability distribution of luminance pixels which is robust all works for all types of images.

Through the result, we found out that some image accuracy has dropped after saturation adjustment for both testset1 and testset2 although the overall accuracy has increased. This is due to over-saturation of image as it creates unnatural color spill-over. However, majority of image still has improved accuracy because saturation enhancement made them more vivid and vibrant without disturbing the natural essence of the image. The factor that needs attention here is saturation factor. We can’t use trial and error as before to determine a fixed value for a group of images. Instead each image needs a different saturation factor depending on the overall colorfulness of the image. Hence, we can first calculate the colorfulness of image based on method proposed in **[2]** and set a suitable threshold for saturation factor. Through this method, we can avoid over-saturation and unnatural color spill over and further increase the similarity score.

Reference

[1] Huang, S. C., Cheng, F. C., & Chiu, Y. S. (2012). Efficient contrast enhancement using adaptive gamma correction with weighting distribution. IEEE transactions on image processing, 22(3), 1032-1041.

[2] Hasler, D., & Suesstrunk, S. E. (2003, June). Measuring colorfulness in natural images. In Human vision and electronic imaging VIII (Vol. 5007, pp. 87-95). International Society for Optics and Photonics.