California State University
Department of Computer Science

CSCI-290

Independent Study Report

On

Image Processing Techniques

(fall 2017)

DATE: 12/15/2017

Submitted by:

Rishabh Sharma

109874160

Submitted to:

Dr. Hubert Cecotti

ABSTRACT

Image quality assessment is important image processing task, especially for the application in which the image is to be seen by the human beings. As we capture image from a digital camera, there is always some random variation in intensity, loss in illumination, and changes in contrast in image due to processing and compression performed by the camera while capturing the image. Transferring, storing or reproduction of the image adds to the level of distortion. Thus, for better quality assessment of the image it is important to preprocess images for these distortions.

In this study, I performed various image modifying and enhancement techniques using linear filters for preprocessing images to overcome the changes in contrast and illumination in the image. I implemented various image quality assessment techniques to understand the impact of the different types distortion on different quality assessment algorithms. For better understanding and comparison, quality assessment algorithm is executed across TID2013 database with and without applying different filters.

ACKNOWLEDGMENTS

I wish to thank my mentor, Dr. Hubert Cecotti for his great teaching, support, and guidance throughout my course-work. Through his guidance and proactive teaching skills, I was able to understand the theoretical and practical aspect of Image Processing. I am thankful to all the faculty of Computer Science dept. for being supportive throughout this beautiful journey. Finally, I would like to extend my heartfelt gratitude to my parents, and my friends for their love, support, and encouragement while pursuing my course of study.

LIST OF CONTENTS

TOPICS *PAGE NO.*

LIST OF THE FIGURES *PAGE NO.*

1. INTRODUCTION: IMAGE PROCESSING

Image processing is a method to perform some operations on an image, in order to get an enhanced image or to extract some useful information from it. It is a type of signal dispensation in which input is image, like video frame or photograph and output may be image or characteristics associated with that image. Usually Image Processing system includes treating images as two-dimensional signals while applying already set signal processing methods to them. [1]

There exist two main methods of Image Processing

1. Analog Image Processing
2. Digital Image Processing

1.1 ANALOG IMAGE PROCESSING

Analog Image Processing refers to the alteration of image through electrical means. The most common example is the television image. The television signal is a voltage level which varies in amplitude to represent brightness through the image. By electrically varying the signal, the displayed image appearance is altered. The brightness and contrast controls on a TV set serve to adjust the amplitude and reference of the video signal, resulting in the brightening, darkening and alteration of the brightness range of the displayed image

1.2 DIGITAL IMAGE PROCESSING

Digital image processing is performed on digital images which is represented as a three – dimensional function, *f (x, y)*, where x and y is the spatial (plane) coordinates, and the amplitude of '*f*' at any pair of coordinates *(x, y)* is called the intensity of the image (r (red), g(green), b(blue) value) at that point [2].

$$f(x, y) = \begin{bmatrix} r(x,y) \\ g(x,y) \\ b(x,y) \end{bmatrix} \quad (1)$$

A digital image is composed of a finite number of elements, each of which has a particular location and value. These element is called picture elements, image elements, and pixels. Pixel is the term used most widely to denote the elements of a digital image. The value of Pixel varies from 0 to 255 for r, g and b in the image. [2]

Digital image processing is performed on the greyscale images. I implemented two out of seven greyscale conversion techniques mentioned in [3-4] for converting image from RGB of N number of rows and M number of columns to grey-scale image *g (x, y).* Figure 1 shows the greyscale image (Average and Human Eye) conversion of the RGB image (Original image)

i. Averaging

$$g(x, y) = (r(x,y) + g(x,y) + b(x,y))/3 \quad (2)$$

ii. Correcting for human eye (sometimes called "luma" or "luminance")

Photoshop/GIMP

$$g(x, y) = r(x,y) * 0.3 + g(x,y) * 0.59 + b(x,y) * 0.11 \quad (3)$$

BT.709 Specification

$$g(x, y) = r(x,y) * 0.2126 + g(x,y) * 0.7152 + b(x,y) * 0.0722 \quad (4)$$

BT.601 Specification

$$g(x, y) = r(x,y) * 0.299 + g(x,y) * 0.587 + b(x,y) * 0.114 \quad (5)$$

*Figure 1. Conversion of RGB image **(left)** to Greyscale using Average **(right top)** and Human Eye technique (using Photoshop/GIMP equation) **(right bottom).***

2. METHODS

Filtering is a technique for modifying or enhancing an image. It is useful in many applications including smoothing, sharping, removing noise, and edge detection. A filter is defined by a kernel, which is a small array (1D or 2D depends upon the application) applied to each pixel, and is a square with an odd number (3, 5, 7, etc.) of each element in each dimension. The process used to apply filters to an image is known as convolution, and may be applied in either spatial and frequency domain. [5]

2.1 SMOOTHING

Gaussian filtering is used to blur images and remove noise and details. In one dimension, the Gaussian function is:

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}} \tag{6}$$

where σ is the standard deviation of the distribution. The distribution is assumed to have a mean of 0. The Gaussian function for two dimensional images given by the equation 7 which is the product of two 1D Gaussian functions (one for each direction).

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{7}$$

The discretize value of the continuous Gaussian function in term of pixel values in 5 by 5 convolution kernel is:

$$\frac{1}{273}\begin{bmatrix}1 & 4 & 7 & 4 & 1\\ 4 & 16 & 26 & 16 & 4\\ 7 & 26 & 41 & 26 & 7\\ 4 & 16 & 26 & 16 & 4\\ 1 & 4 & 7 & 4 & 1\end{bmatrix} \quad (8)$$

The reason for the blurry effect using Gaussian kernel is due to the fact the Gaussian kernel coefficient diminish with increasing distance from the kernel's center which gives the central pixel higher weighing than those on the periphery I used Gaussian blur because it is an edge preserving blur and it has been found that neurons in human brain create a similar filter when processing visual images. [6] Figure 2 shows the implementation of the Gaussian blur on the grey-scale image (calculated using average).

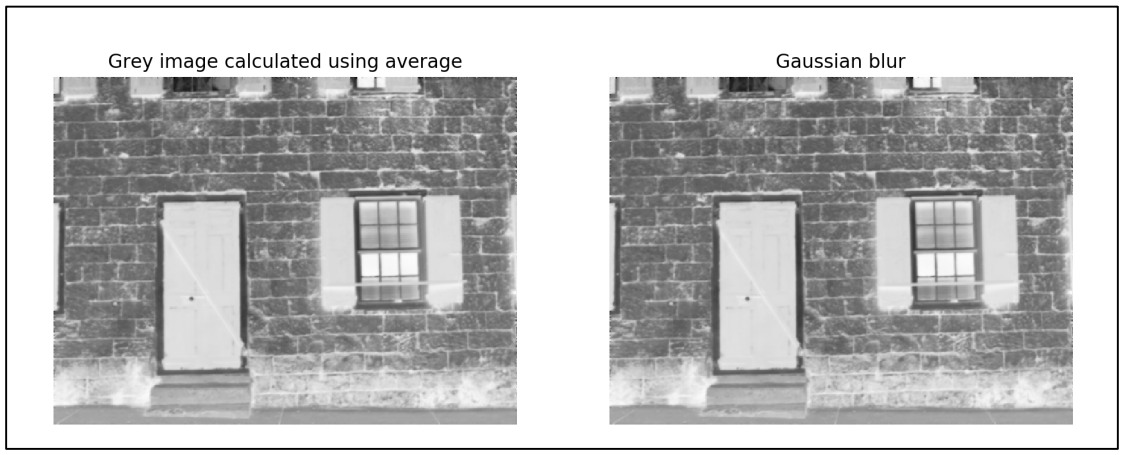

*Figure 2. Conversion of the image form grey image **(left)** to Gaussian blur **(right).***

2.2 BINARIZATION

Binarization is the technique of converting the image from grey scale to black and white. I implemented Otsu's algorithm for performing binarization. Otsu (named after Nobuyuki Otsu) is a global image thresholding algorithm usually used for

thresholding, binarization and segmentation. It works mainly with the image histogram, looking at the pixel values and the regions that the user wants to segment out, rather than looking at the edges of an image. It tries to segment the image making the variance on each of the classes minimal. The algorithmic implementation of the Otsu method is referred from [7]. Figure 3 shows the implementation of the Otsu binarization on the Gaussian blur image.

*Figure 3. Conversion of the image form Gaussian blur **(left)** to Otsu Binarization **(right).***

Algorithm 1: Thresholding segmentation using Otsu's method or manual input

```
input  : input_image (grayscale), overridden_threshold
output: output_image
read(input_image)
N = input_image.width × input_imnumberage.height
                                                    initialize variables
threshold, var_max, sum, sumB, q1, q2, μ1, μ2 = 0
max_intensity = 255
for i=0; i <= max_intensity; i++ do
    histogram[value] = 0
                                                    accept only grayscale images
if num_channels(input_image) > 1 then
    return error
                                                    compute the image histogram
for i=0; i<N; i++ do
    value = input_image[i]
    histogram[value] += 1
if manual threshold was entered then
    threshold = overridden_threshold
else
                                                    auxiliary value for computing μ2
    for i=0; i<=max_intensity; i++ do
        sum += i × histogram[i]
                                                    update qi(t)
    for t=0; t<=max_intensity; t++ do
        q1 += histogram[t]
        if q1 == 0 then
            continue
        q2 = N − q1
                                                    update μi(t)
        sumB += t × histogram[t]
        μ1 = sumB/q1
        μ2 = (sum - sumB)/q2
                                                    update the between-class variance
        σb²(t) = q1(t)q2(t)[μ1(t) − μ2(t)]²
                                                    update the threshold
        if σb²(t) > var_max then
            threshold = t
            var_max = σb²(t)
                                                    build the segmented image
for i=0; i < N; i++ do
    if  input_image[i] > threshold then
        output_image[i] = 1
    else
        output_image[i] = 0
return output_image
```

Figure 4. Otsu Binarization Algorithm [7].

2.3 EDGE DETECTION

Edge detection is one of the important image filtering techniques. Edge detection refers to the process of identifying and locating sharp discontinuities in an image. The discontinuities are abrupt changes in pixel intensity which characterize boundaries of object in a scene. Classical methods of edge detection involve convolving the image with an operator (a 2-D filter), which is constructed to be sensitive to large gradients in the image while returning values of zero in uniform regions. There are an extremely large number of edge detection operators available, each designed to be sensitive to certain types of edges. [8]

SOBEL OPERATOR

Most edge detection methods work on the assumption that the edge occurs where there is a discontinuity in the intensity function or a steep intensity gradient in the image. Using this assumption, if one takes the derivative of the intensity value across the image and find points where the derivative is maximum, then the edge could be located. The gradient is a vector, whose components measure how rapid pixel value are changing with distance in the x and y direction. Thus, the components of the gradient may be found using the following approximation: [9] Figure 4. shows the changes in the Sobel edge detected with and without performing the Gaussian blur

Pseudo-codes for Sobel edge detection method

Input: A Sample Image
Output: Detected Edges
Step 1: Accept the input image
Step 2: Apply mask Gx, Gy to the input image
Step 3: Apply Sobel edge detection algorithm and the gradient
Step 4: Masks manipulation of Gx, Gy separately on the input image
Step 5: Results combined to find the absolute magnitude of the gradient

$$|G| = \sqrt{Gx^2 + Gy^2} \tag{9}$$

Step 6: the absolute magnitude is the output edges

Filtering the original image following two kernels gives the result for Gx and Gy. [9]

$$Kernel\ 1 = Gx = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 1 & 2 \\ -1 & 0 & 1 \end{bmatrix} \tag{10}$$

$$Kernel\ 2 = Gy = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{11}$$

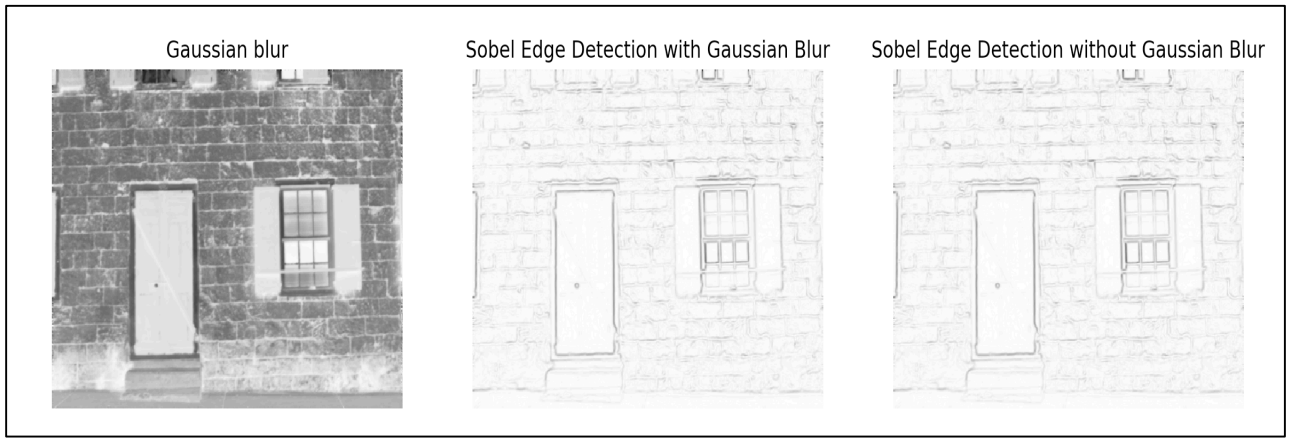

Figure 5. Sobel Edge Detection with and without Gaussian filters.

*Gaussian Blur **(left)** edge detection with Gaussian Blur **(middle)** edge detection without Gaussian Blur **(right).***

2.4 CONNECT COMPONENT IN IMAGES

Connected-component labeling is a procedure for assigning a unique label to each object (or a connected component) in an image. Because these labels are key for other analytical procedures, connected-component labeling is an indispensable part of most applications in pattern recognition and computer vision, such as character recognition. In many cases, it is one of the most time-consuming tasks among other pattern-recognition algorithms. Therefore, connected-component labeling continues to be an active area of research. [10]

I implemented a connect component labeling algorithm mentioned in [10]. The input for the algorithm is the segmented image (binary image) obtained from the Otsu algorithm. The binary image contains two type of pixels object pixels (255) and background pixels (0). The labeling task is performed on the object pixels such that each object pixel that all the connected object pixels (neighboring object pixels) are having the same labels. The paper [10] has implemented 8 – connectedness as shown in figure 6 for finding the connectivity among the neighbors of the pixel ‘e’. For every pixel ‘e’ in the image the pixel ‘a’, ‘b’, ‘c’, and ‘d’ are designated as scan mask pixels.

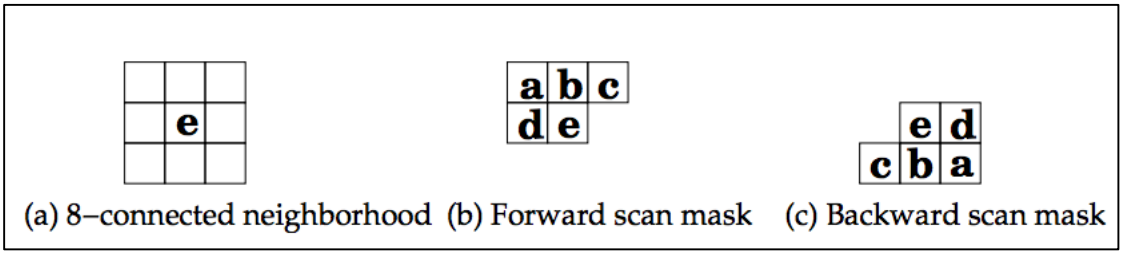

Figure 6. The masks and the neighborhood of pixel ‘e’.
*8 – connected neighborhood **(left)** Forward scan mask **(middle)** Backward can mask **(right).***

If there is no object pixel present in the scan mask, the current pixel receives a new provisional label. If on the other hand, the object pixel is present in the scan mask, the provisional label of the neighbor is considered equivalent, a representative label is selected to represent all equivalent labels, and the current object pixel is assigned this representative label. A common strategy for selecting a representative is to use the

smallest label. [10] Figure 7 shows the implementation of the connect component algorithm on the Otsu image.

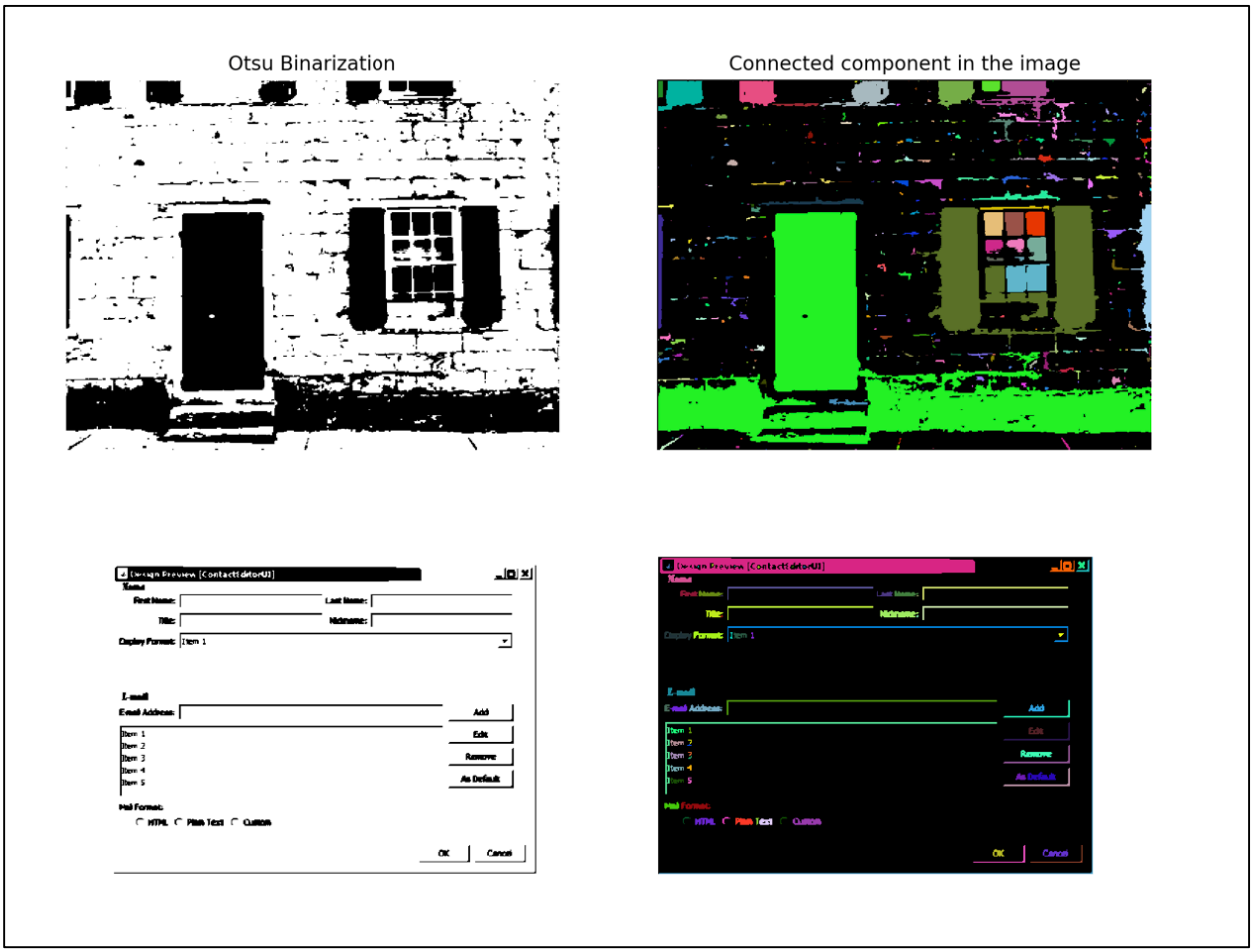

Figure 7. Connected components represented by different colors.
Otsu Binarization of image from TID2013 ***(top left)*** *connected component of image from TID2013* ***(top right).***
Otsu Binarization of GUI ***(bottom left)*** *connected component of GUI* ***(bottom right).***

3. DISTANCE

3.1 STRUCTURAL SIMILARITY INDEX

Structural similarity is the quantitative measurement for checking quality of the image. Image quality can be measured in different ways using full – reference, reduced – referenced and no – referenced methods. [11] Structural similarity method for image quality estimation is implemented using full – referenced techniques. Structural similarity Index (SSIM) is the quality measure from the prospective of image formation. To explore the structural information of the image, factors like luminance measurement, contrast comparison and structure comparison are taken into consideration.

For measuring the structural similarity between two images x and y using SSIM luminance *l (x, y)*, contrast comparison *c (x, y)* and structure comparison *s (x, y)* is calculated first using equation 12, 13 and 14 respectively. Where constant $C_1, C_2,$ and C_3 is added to avoid instability when denominator is close to zero. Whereas μ_x and σ_x represent the mean and standard deviation of the image x respectively and similarly μ_y and σ_y represent the mean and standard deviation of the image y and are given by equation 15 and 16.

$$l(x,y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \tag{12}$$

$$c(x,y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \tag{13}$$

$$s(x,y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \tag{14}$$

$$\mu_x = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{15}$$

$$\sigma_x = \left(\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu_i)^2\right)^{\frac{1}{2}} \quad (16)$$

Finally, all the three factors are combined into one to calculate the SSIM values using equation 17 and figure 8 shows the value of SSIM on 24 distortions given in Tampere Image Database 2013 (TID2013) over the worst level (i.e., level 5).

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (17)$$

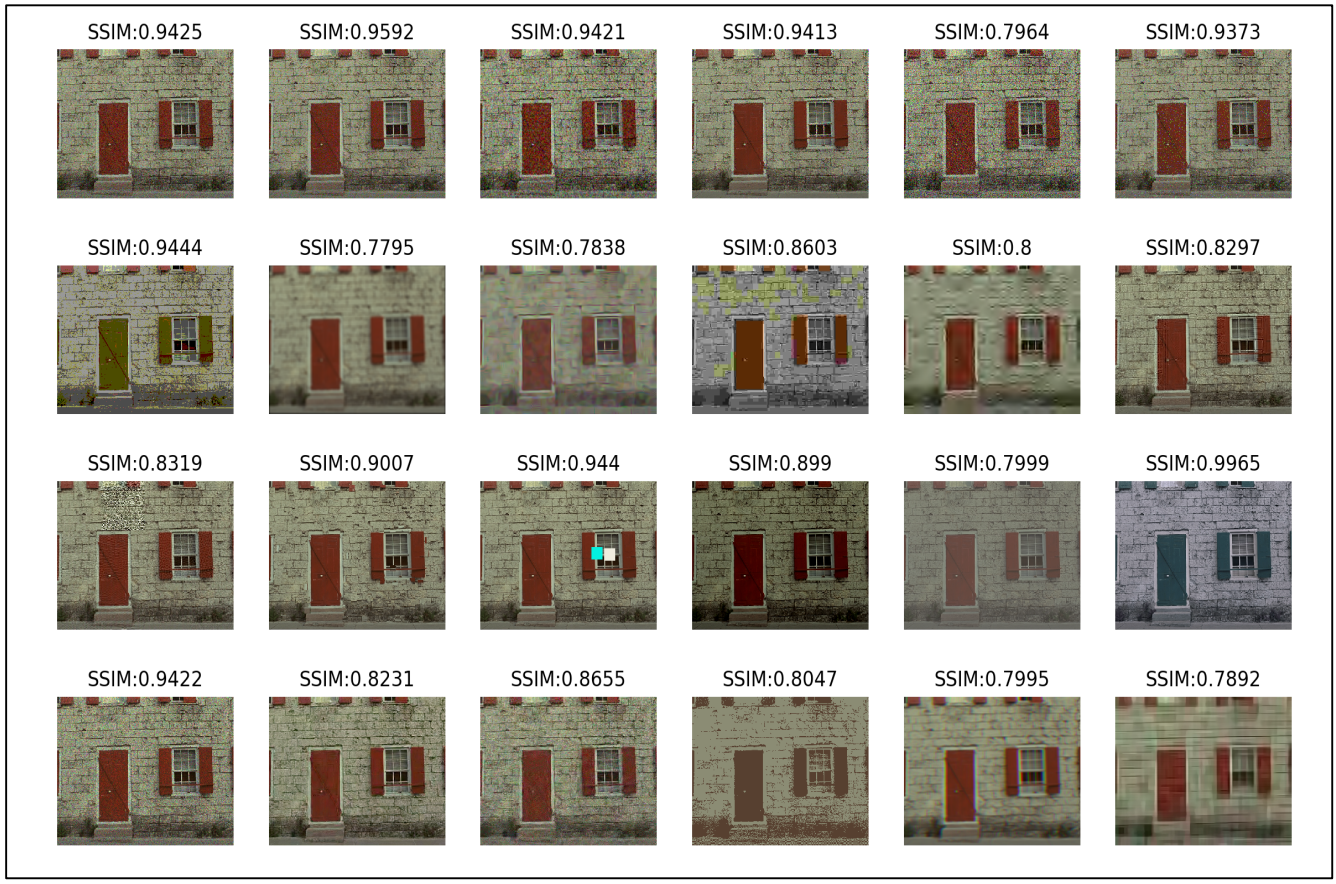

Figure 8. SSIM value of the 24-different distortion of level 5 on the image database TID2013. Image with distortion 1 ***(top left)*** *to the distortion 24* ***(bottom right).*** *One means highly similar and zero means completely different.*

3.2 ROOT MEAN SQUARE

Root Mean Square (RMS) is a measure to compare two signals by providing a quantitative score describing the degree of similarity or the level of distortion between them. Let the two images x and y of size N, where N is the total number of pixels present in the image. The RMS value of the comparison between two images x and y is given by equation 18 and figure 9 shows the value of RMS on different types of distortion in TID2013.

$$MSE(x, y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2 \tag{18}$$

Figure 9. RMS value of the 24-different distortion of level 5 on the image database TID2013. Image with distortion 1 ***(top left)*** *to the distortion 24* ***(bottom right).*** *One means highly similar and zero means completely different.*

3.3 STRUCTURAL SIMILARITY INDEX – WINDOW SPLIT

I performed Structural Similarity Index (SSMI) algorithm by splitting the original image and the image to be compare into four equal half (A, B, C, D) each as shown in figure 10 and comparing A, B, C, and D region of image one with A, B, C, and D region of image two. Figure 11 shows the value of SSIM with four windows on different types of distortion in TID2013.

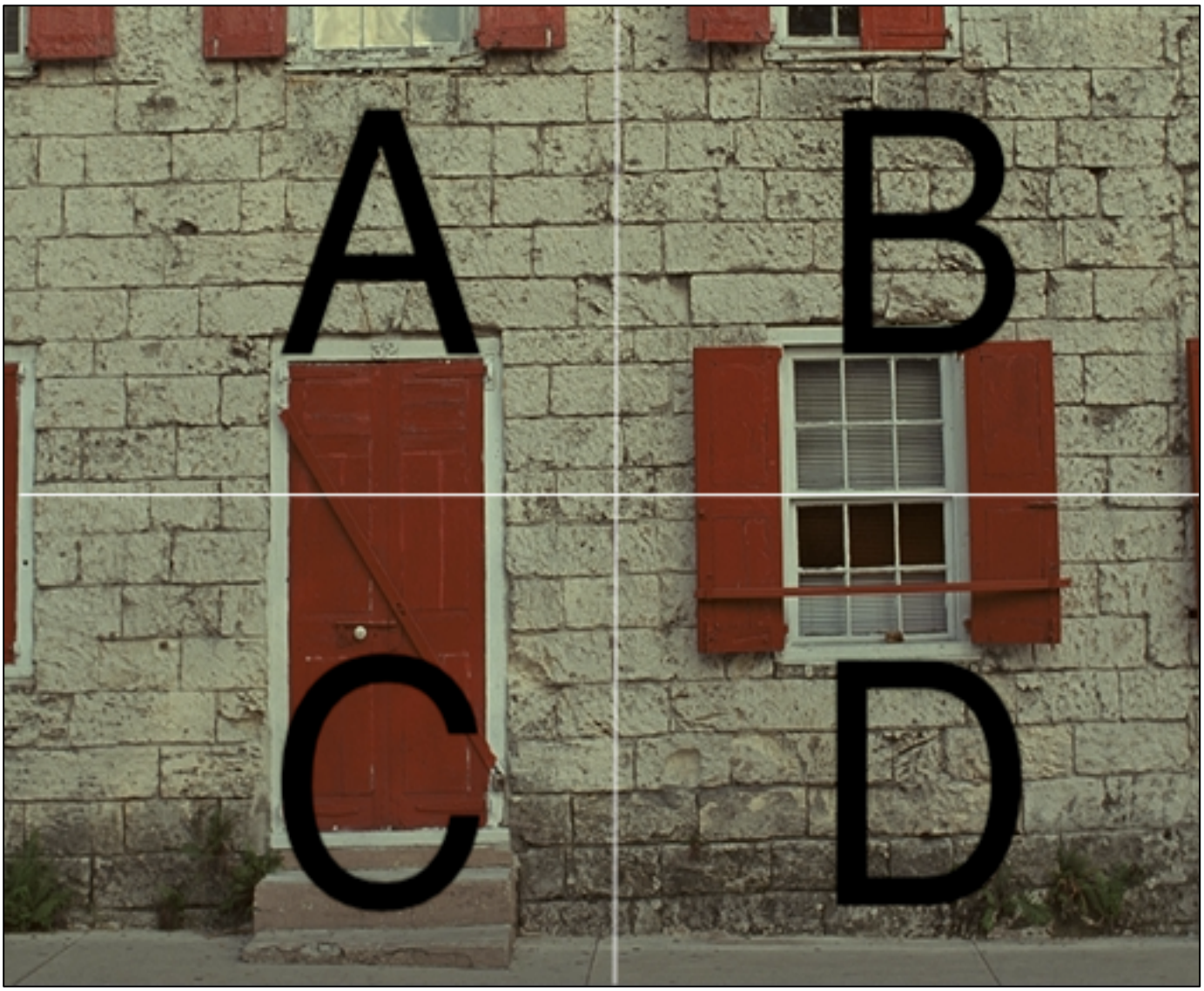

*Figure 10. Image split into 4 equal half A **(top left)**, B **(top right)**, C **(bottom left)** and D **(bottom right)**.*

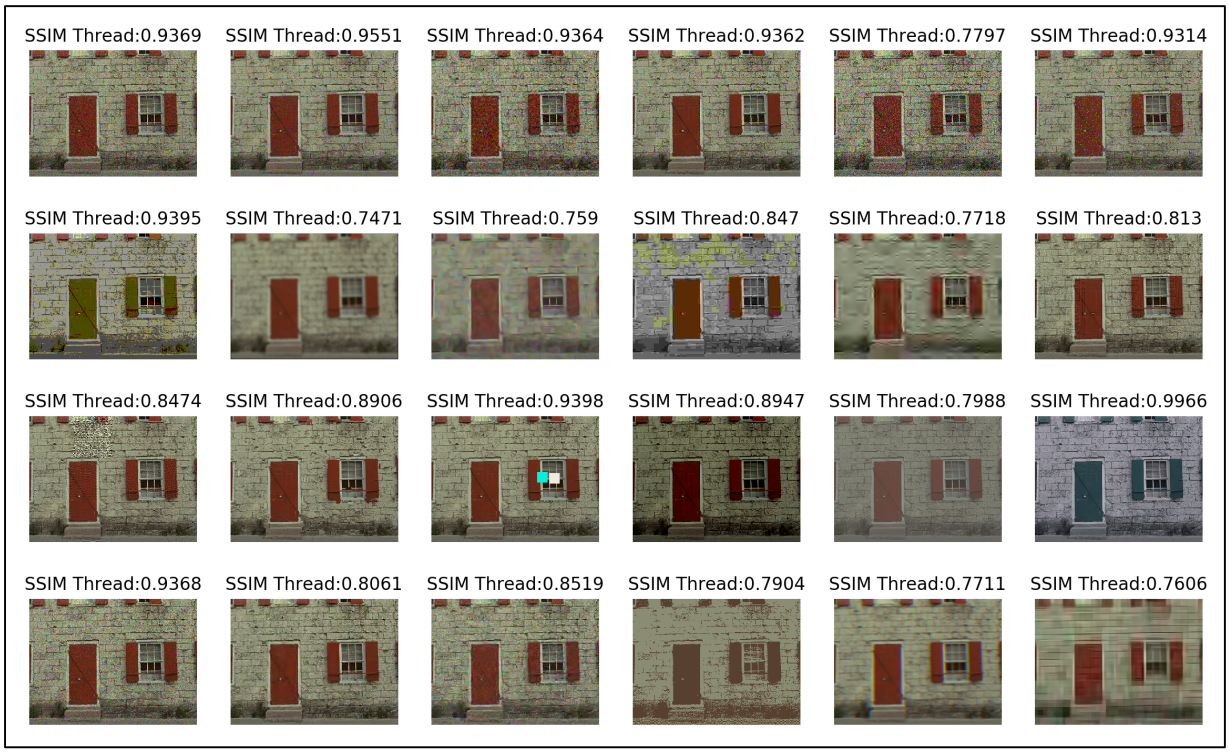

Figure 11. SSIM - Window value of the 24-different distortion of level 5 on the image database TID2013.
Image with distortion 1 ***(top left)*** *to the distortion 24* ***(bottom right).***
One means highly similar and zero means completely different.

4. RESULTS

4.1 TAMPERE IMAGE DATABASE

The above mentioned 3 distances SSIM, SSIM – Window, and RMS are calculated on Tampere Image database 2013 (TID2013). TID2013 is an image database containing 25 reference images and 3000 distorted images (25 reference images x 24 types of distortion x 5 levels of distortions). All the images are in bitmap image format. [13, 14, 15] All images were of the same fixed size 512 x 384 figure 12 shows the reference of the images opresent in the TID2013.

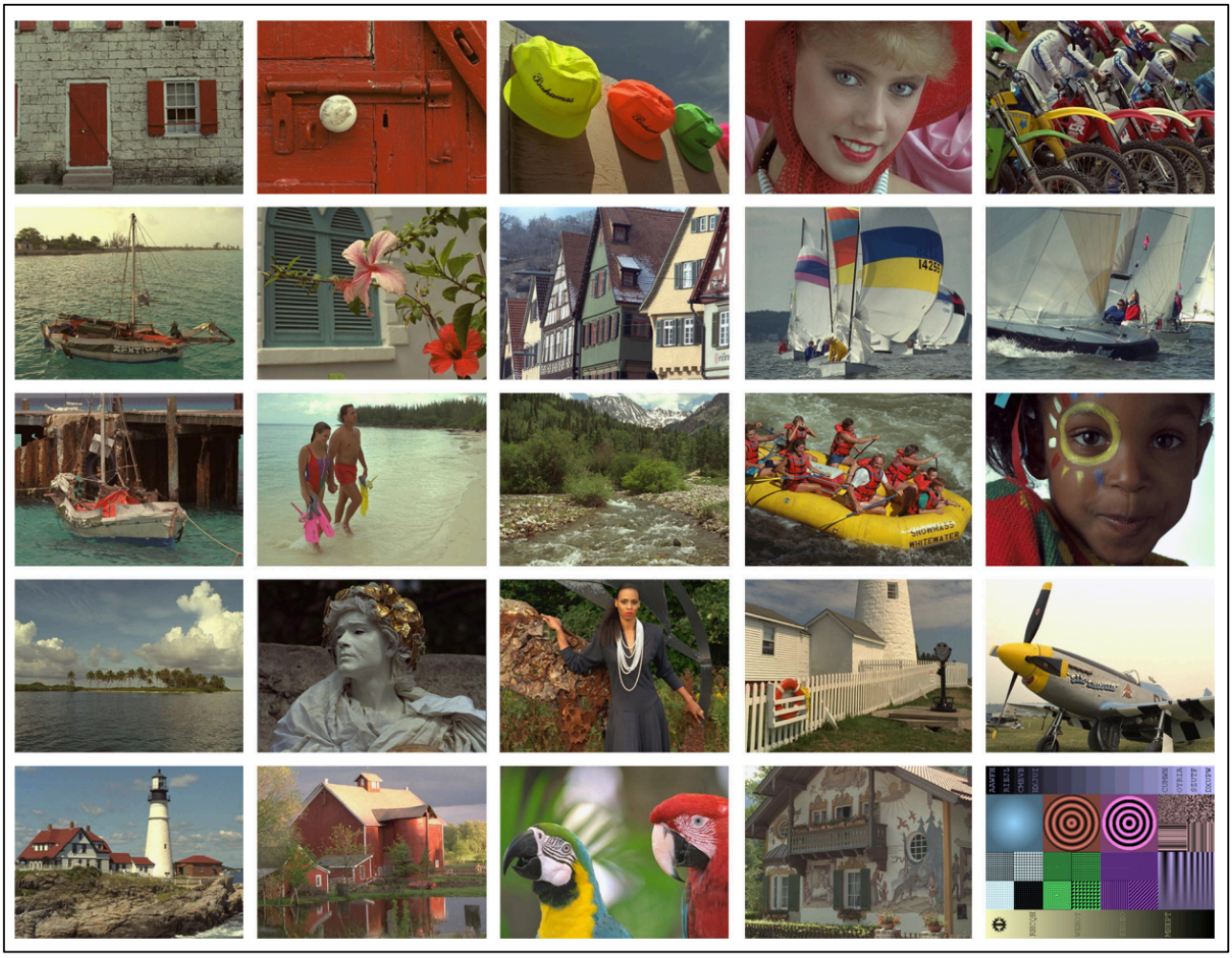

Figure 12. Reference images in TID2013

Table 1. TID2013: twenty-four types of distortion

N	Type of distortion
1	Additive Gaussian noise
2	Additive noise in color components is more intensive than additive noise in the luminance component
3	Spatially correlated noise
4	Masked noise
5	High frequency noise
6	Impulse noise
7	Quantization noise
8	Gaussian blur
9	Image denoising
10	JPEG compression
11	JPEG2000 compression
12	JPEG transmission errors
13	JPEG2000 transmission errors
14	Non-eccentricity pattern noise
15	Local block-wise distortions of different intensity
16	Mean shift (intensity shift)
17	Contrast change
18	Change of color saturation
19	Multiplicative Gaussian noise
20	Comfort noise
21	Lossy compression of noisy images
22	Image color quantization with dither
23	Chromatic aberrations
24	Sparse sampling and reconstruction

4.2 RESULT ON TID2013 WITHOUT FILTERING

Figure 13, 14, 15, 16 and 17 shows the calculation of the SSIM, SSIM- Window split and RMS method for measuring the similarity between the images present in TID2013 database. 24 bars represent the 24-different kind of distortion listed in table 1, height of each bar is the average of the similarity measurement over all the 25 images (with distortion) and 25 reference images present in the TID database. Error bar shows the standard deviation value in the similarity measurement for those 25 images.

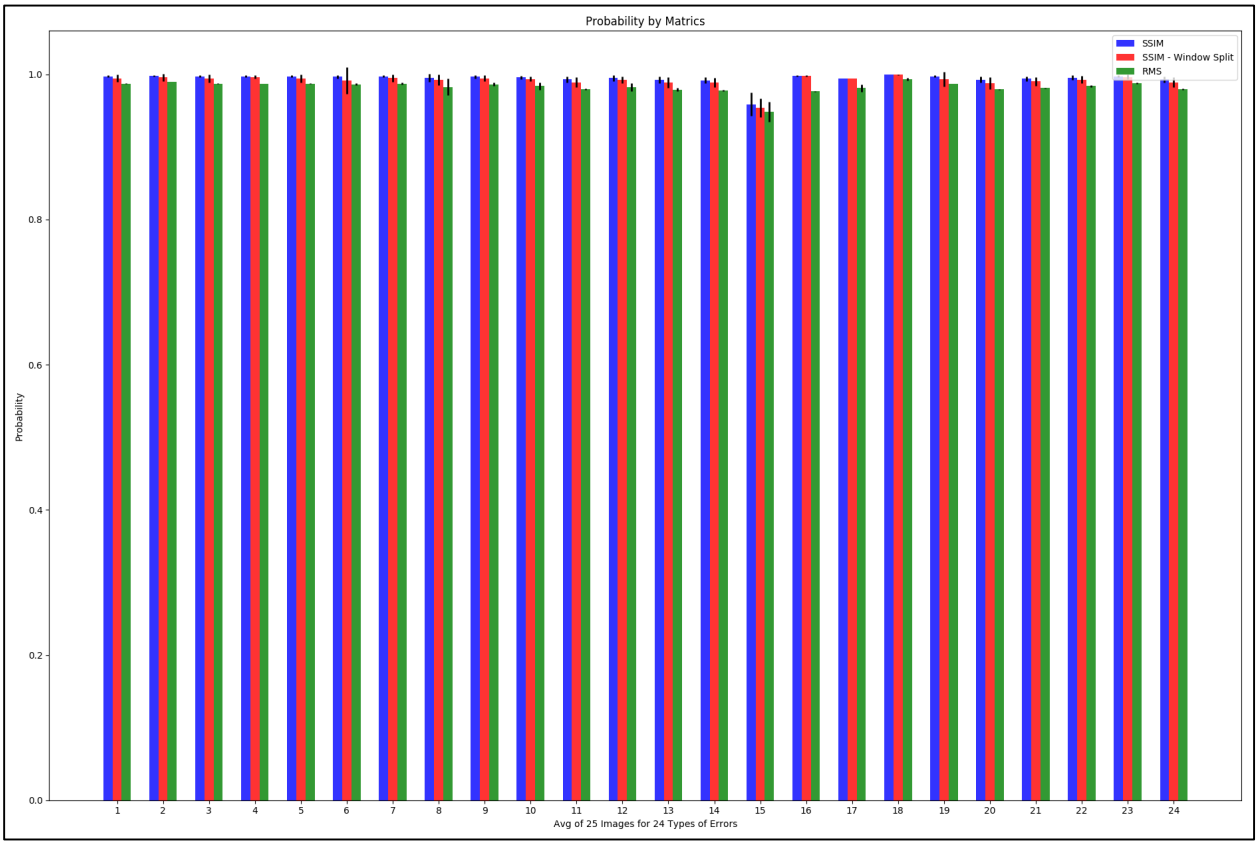

Figure 13. Result for the distortion for type 1

X – axis average of 25 images for 24 types of error

Y – axis similarity measurement of the distorted image with the reference image

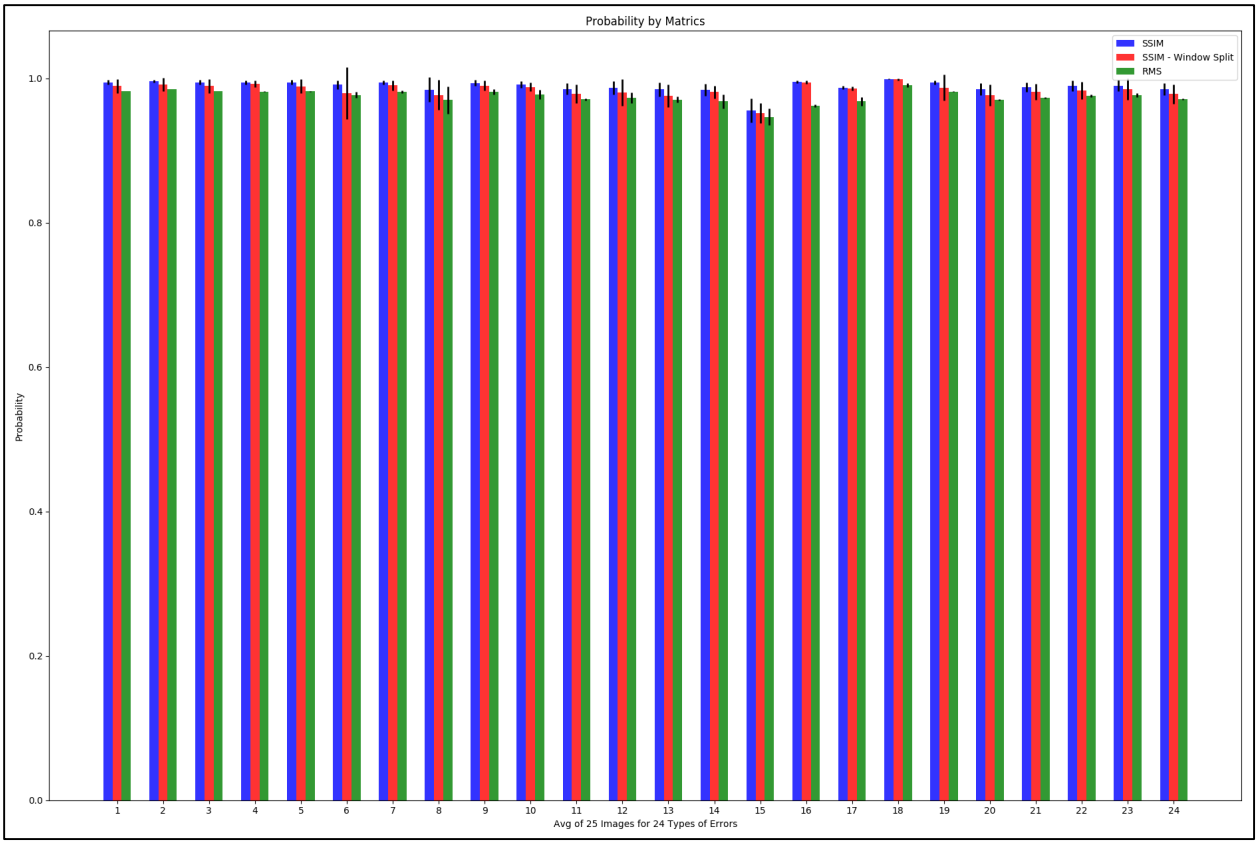

Figure 14. Result for the distortion for type 2

X – axis average of 25 images for 24 types of error

Y – axis similarity measurement of the distorted image with the reference image

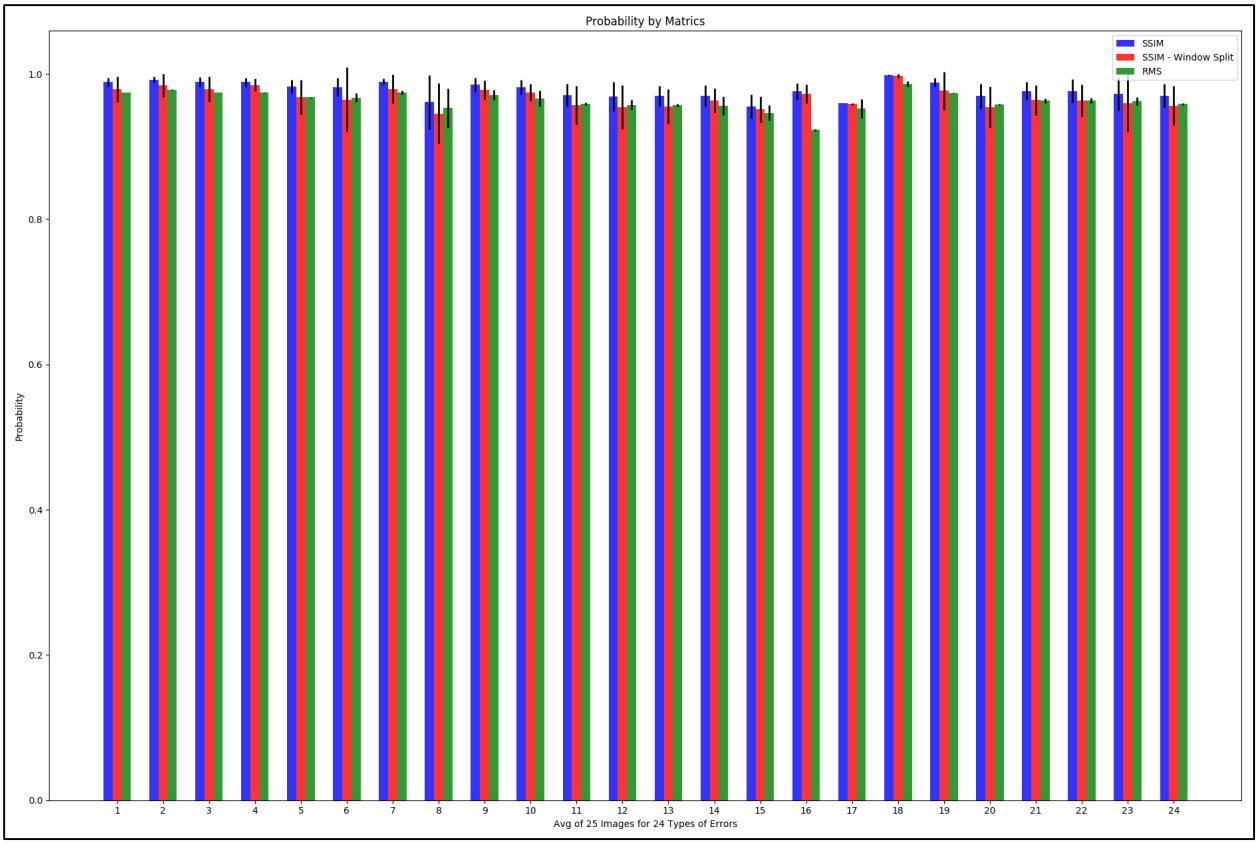

Figure 15. Result for the distortion for type 3

X – axis average of 25 images for 24 types of error

Y – axis similarity measurement of the distorted image with the reference image

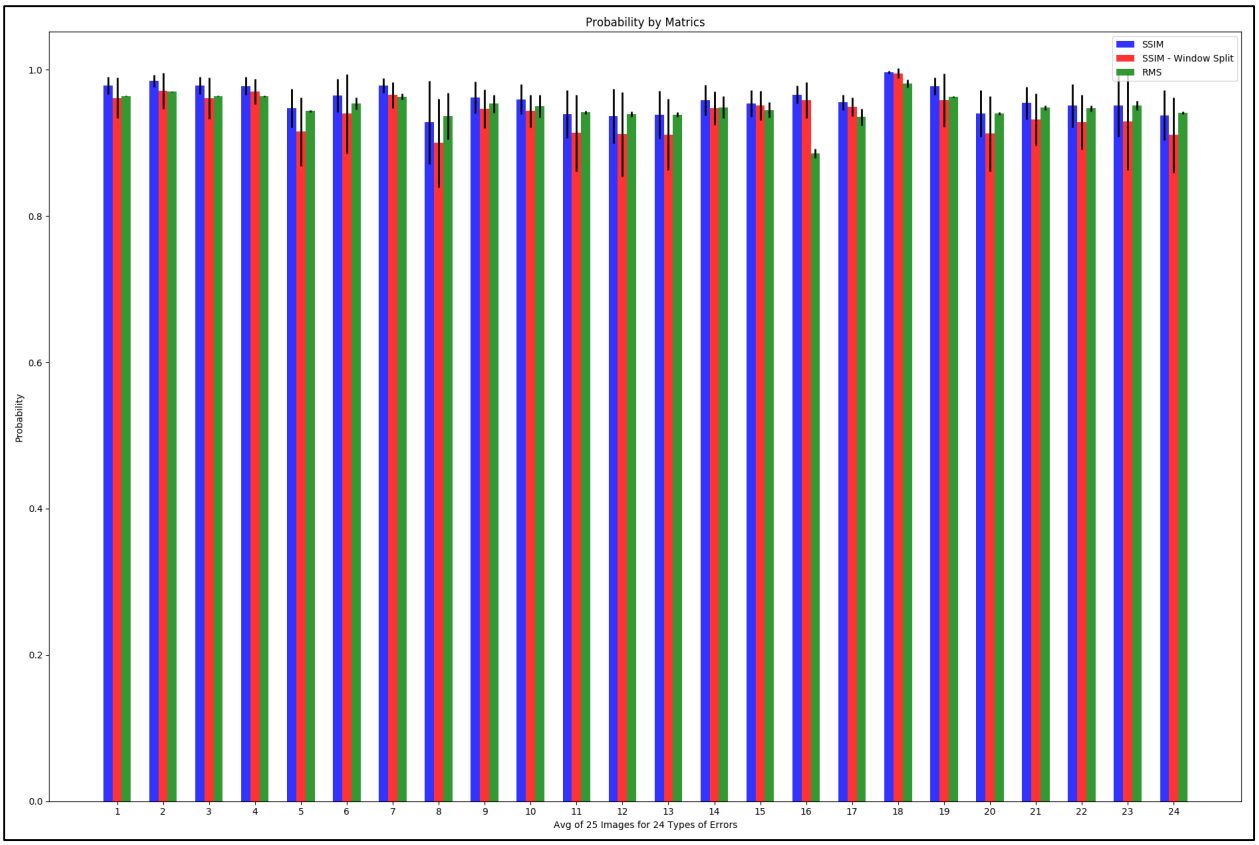

Figure 16. Result for the distortion for type 4

X – axis average of 25 images for 24 types of error

Y – axis similarity measurement of the distorted image with the reference image

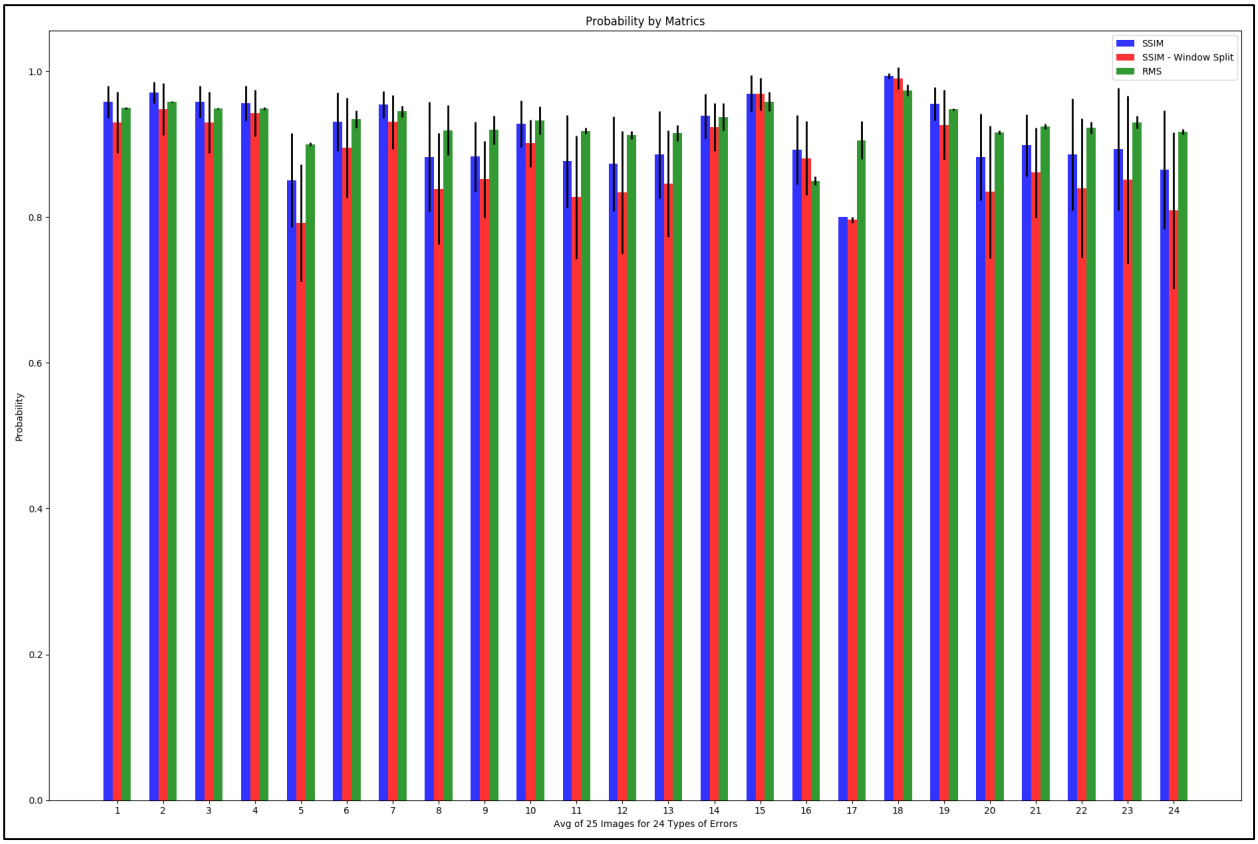

Figure 17. Result for the distortion for type 5

X – axis average of 25 images for 24 types of error

Y – axis similarity measurement of the distorted image with the reference image

4.3 RESULT ON TID2013 AFTER BINARIZATION

Figure 18, 19, 20, 21 and 22 shows the calculation of the SSIM, SSIM- Window split and RMS method for measuring the similarity between the images present in TID2013 database after performing Otsu algorithm on them. 24 bars represent the 24-different kind of distortion listed in table 1, height of each bar is the average of the similarity measurement over all the 25 images (with distortion) and 25 reference images present in the TID database. Error bar shows the standard deviation value in the similarity measurement for those 25 images.

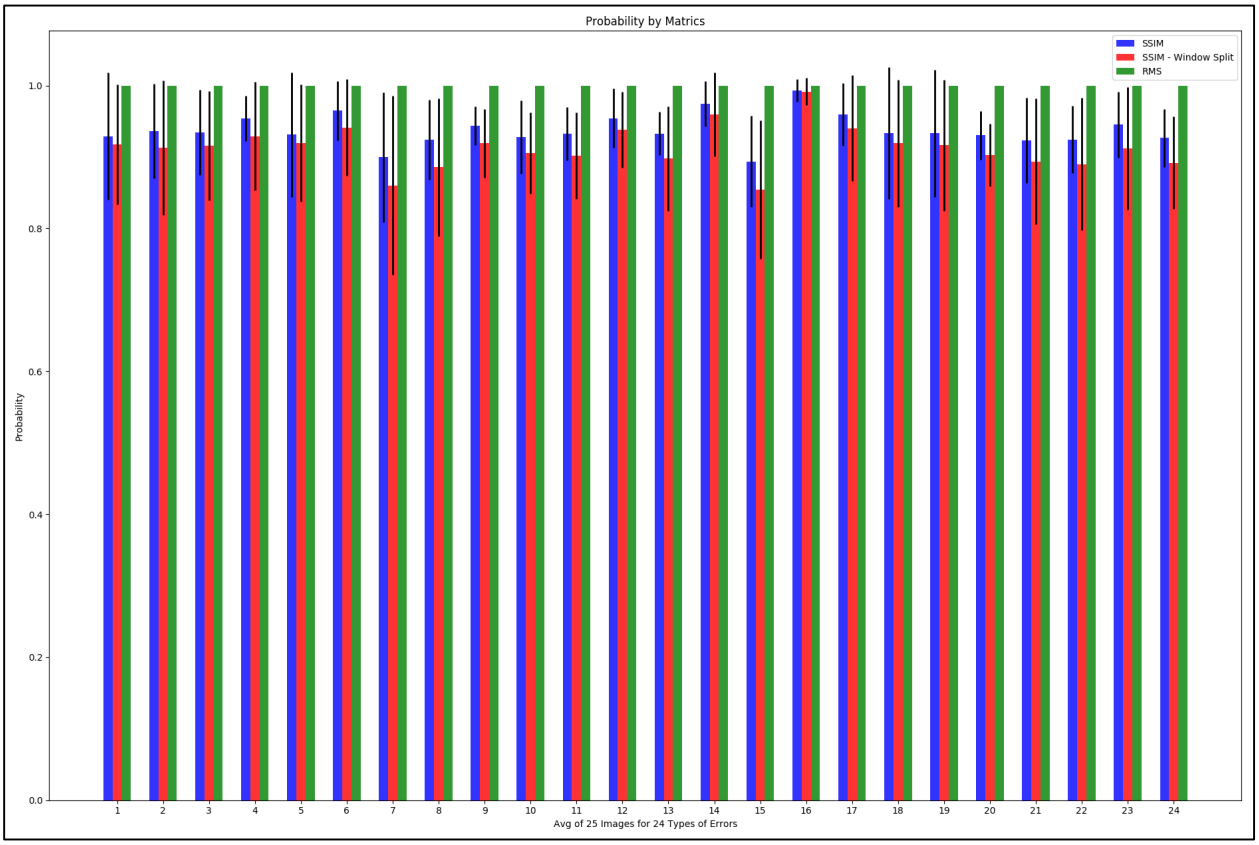

Figure 18. Result for the distortion for type 1 using Otsu filter

X – axis average of 25 images for 24 types of error

Y – axis similarity measurement of the distorted image with the reference image

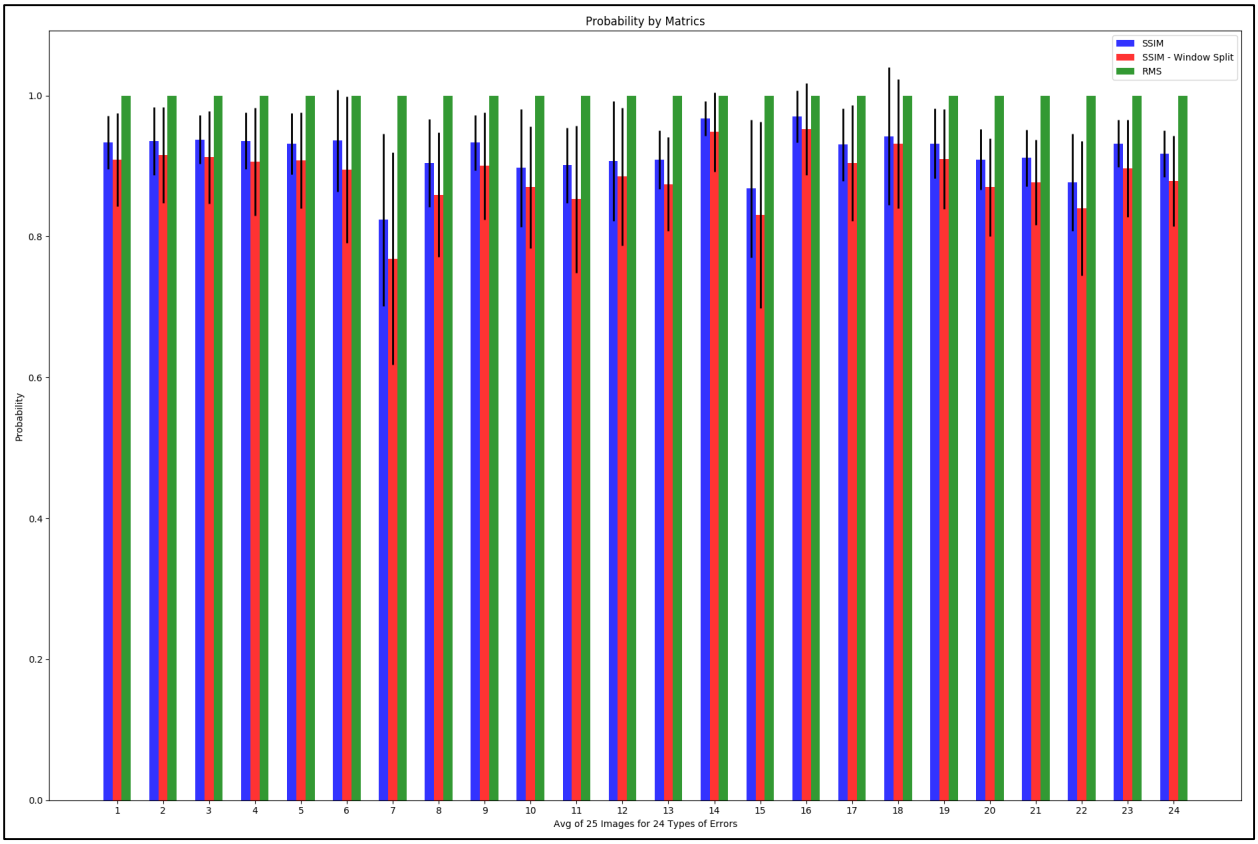

Figure 19. Result for the distortion for type 2 using Otsu filter

X – axis average of 25 images for 24 types of error

Y – axis similarity measurement of the distorted image with the reference image

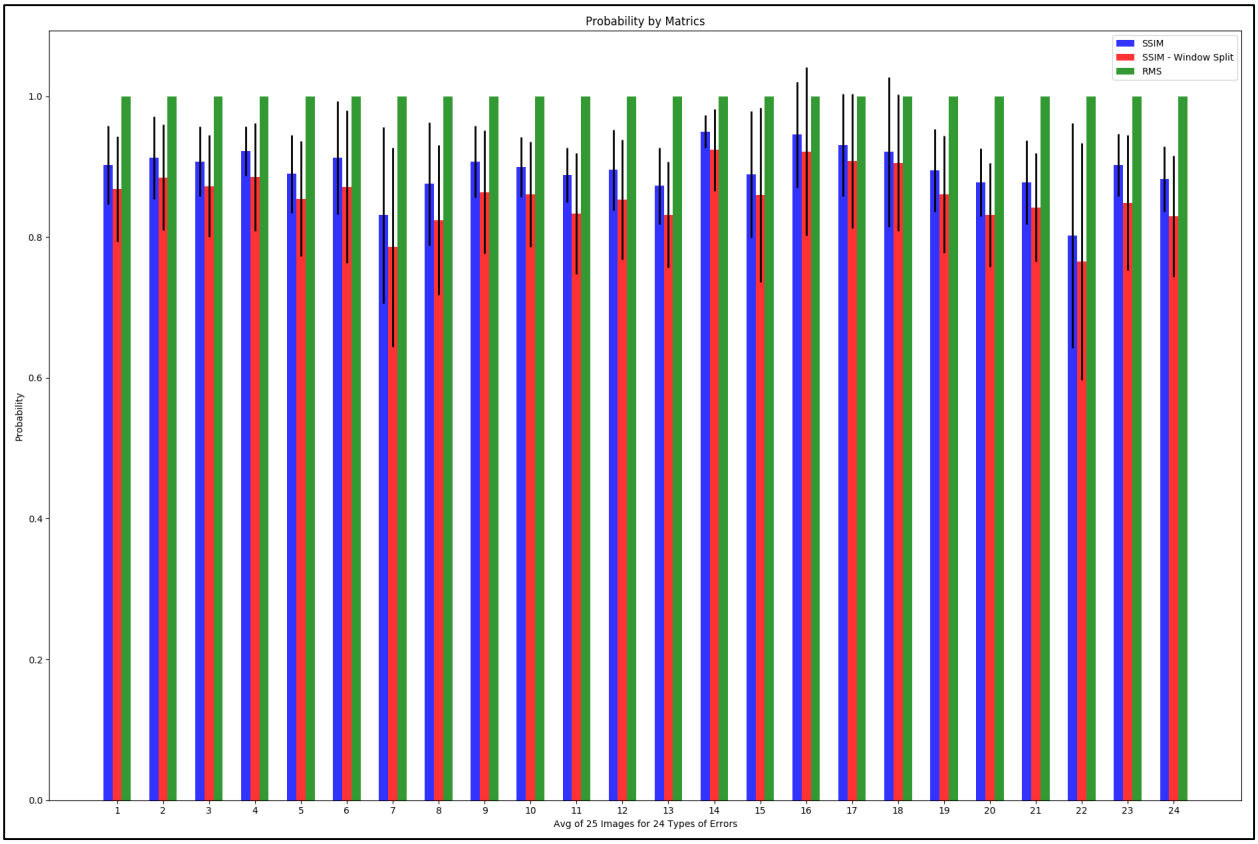

Figure 20. Result for the distortion for type 3 using Otsu filter

X – axis average of 25 images for 24 types of error

Y – axis similarity measurement of the distorted image with the reference image

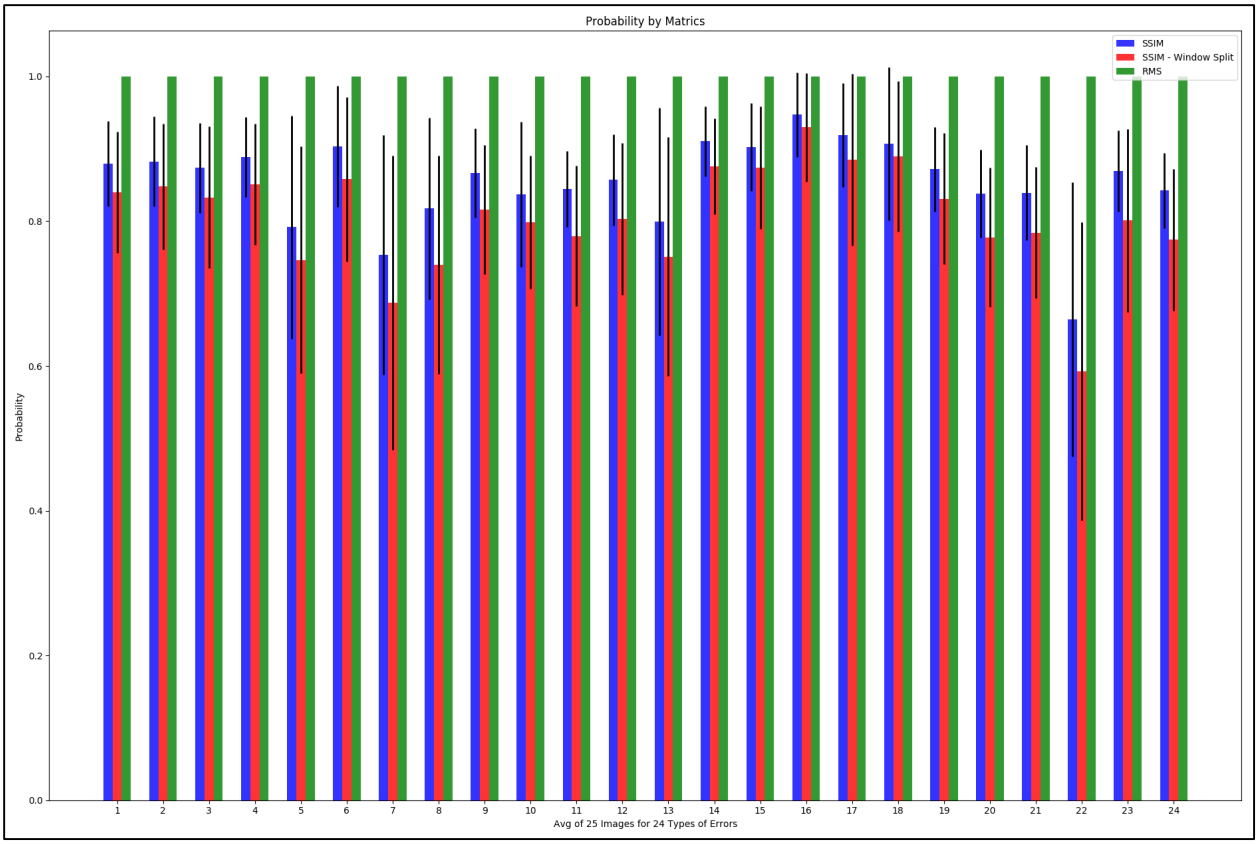

Figure 21. Result for the distortion for type 4 using Otsu filter

X – axis average of 25 images for 24 types of error

Y – axis similarity measurement of the distorted image with the reference image

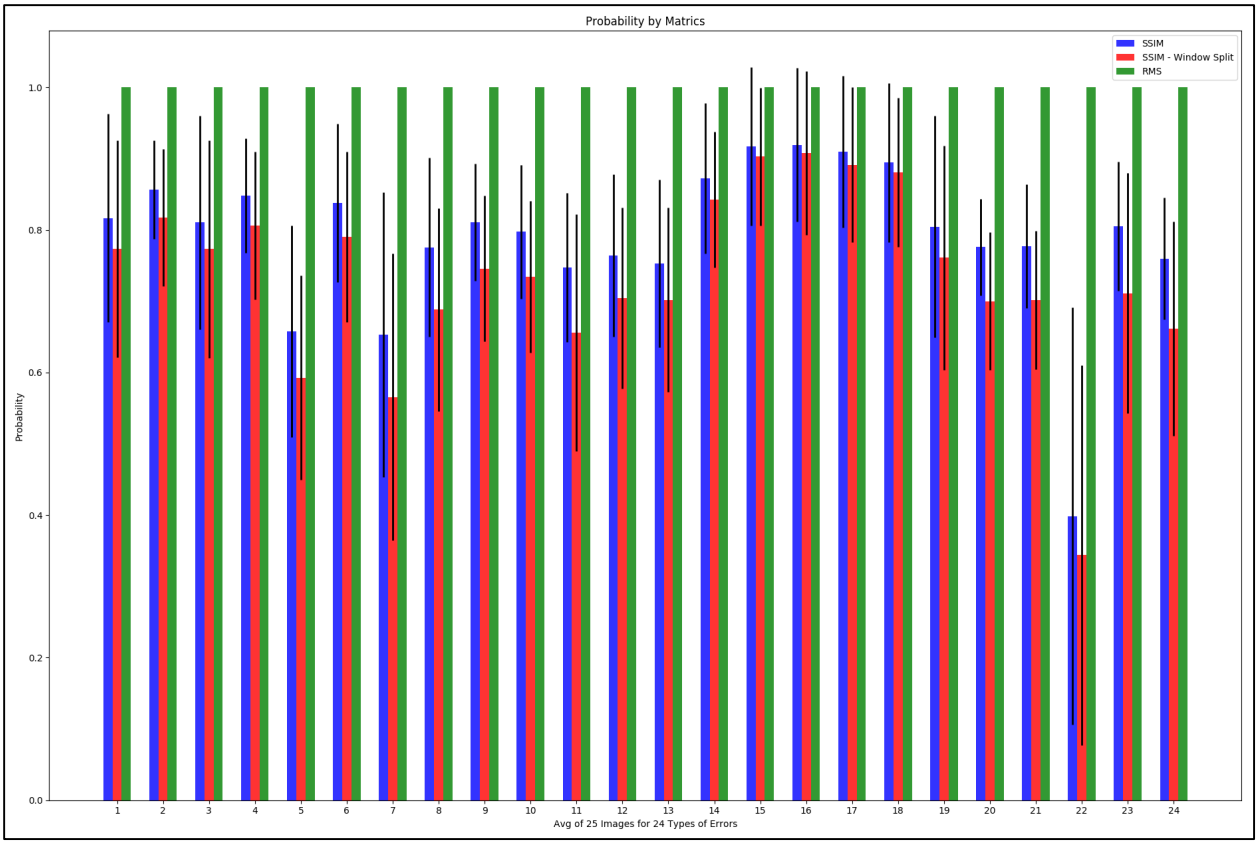

Figure 22. Result for the distortion for type 5 using Otsu filter

X – axis average of 25 images for 24 types of error

Y – axis similarity measurement of the distorted image with the reference image

5. CONCLUSION AND DISCUSSION

In conclusion, I learned different type of image processing filters and algorithms used for linear image filtering like Gaussian filter for smoothening the image, Otsu filters for binarization, and Sobel filters for edge detection. I implemented the two – pass method for finding the connected – components in a binarized image and display each connected component with different images. The connected colored component image is used for identifying the presence of different component present in the image.

I learned the three different similarity measurement techniques SSIM, RMS and SSIM – window and implement these techniques on the TID2013 database which contains 25 different images with 24 distortions for each image and 5 levels of alteration for each type of distortion which sums up to 3000 different distorted images. From the experimented results, I observed that:

For similarity measurement performed without using any filters:

- For majority of the distortions, SSIM results better than SSIM – window and RMS.
- The values of the similarity measurement techniques (SSIM, SSIM – window and RMS) is maximum for the distortion of level 0 where the level of alteration in the distorted image is minimum and it decreases gradually for level 2, level 3, level 4 and for level 5 where the alteration is maximum in distorted image.
- Results for first 4 levels (level 1 to level 4) of alteration shows a pattern where the SSIM results the best (maximum) similarity value and RMS results the worst (minimum) similarity values, while SSIM – window techniques result similarity value close to the SSIM values but always less than SSIM values.
- Results for level 5 when the alteration level is maximum for the distortion in the image, RMS results for the distortion like Contrast change, Multiplicative Gaussian noise, Comfort noise, Lossy compression of noisy images, Image color quantization with dither, Chromatic aberrations, Sparse sampling and reconstruction distortion is better than SSIM and SSIM – window both.

For similarity measurement performed using Otsu:

- Since the both images (original and distorted) are parsed through the binarization filter (Otsu), the factors like luminance, and contrast does not affect the similarity measurement which results in RMS giving the maximum values (1.0) for every type of distortion and at every level of alteration.
- Even after binarization structure of image is having impact on the SSIM and SSIM – window, due to this the similarity measurement of SSIM and SSIM – window is less approximately 20% less in average compare to RMS.

6. REFERENCES

[1] Kavita, Ritika S., B. Rajani, and S. Sunita. "Review paper on Overview of Image Processing and Image Segmentation." International Journal of Research in Computer Applications and Robotics 1.7 (2013): 1-13.

[2] Gonzalez, R., & Woods, R. (2002). Digital image processing (2nd ed.). Upper Saddle River, N.J.: Prentice Hall.

[3] [online] Available: T. Helland, Seven grayscale conversion algorithms (with pseudocode and VB6 source code) http://www.tannerhelland.com/3643/grayscale-imagealgorithm-vb6/

[4] C. Saravanan, "Color Image to Grayscale Image Conversion," 2010 Second International Conference on Computer Engineering and Applications, Bali Island, 2010, pp. 196-199. doi: 10.1109/ICCEA.2010.192

[5] [online] Available: Filtering in Image, http://northstar-www.dartmouth.edu/doc/idl/html_6.2/Filtering_an_Imagehvr.html

[6] [online] Available: https://www.cs.auckland.ac.nz/courses/compsci373s1c/PatricesLectures/Gaussian%20Filtering_lup.pdf.

[7] Balarini, Juan Pablo, and Sergio Nesmachnow. "A C++ implementation of Otsu's image segmentation method." Image Processing On Line 6 (2016): 155-164.

[8] Maini, Raman, and Himanshu Aggarwal. "Study and comparison of various image edge detection techniques." International journal of image processing (IJIP) 3.1 (2009): 1-11.

[9] Vincent, O. Rebecca, and Olusegun Folorunso. "A descriptive algorithm for sobel image edge detection." Proceedings of Informing Science & IT Education Conference (InSITE). Vol. 40. 2009.

[10] Wu, Kesheng, Ekow Otoo, and Kenji Suzuki. "Optimizing two-pass connected-component labeling algorithms." Pattern Analysis and Applications 12.2 (2009): 117-135.

[11] Wang, Zhou, et al. "Image quality assessment: from error visibility to structural similarity." IEEE transactions on image processing 13.4 (2004): 600-612.

[12] Wang, Zhou, and Alan C. Bovik. "Mean squared error: Love it or leave it? A new look at signal fidelity measures." IEEE signal processing magazine 26.1 (2009): 98-117.

[13] Ponomarenko, Nikolay, et al. "Image database TID2013: Peculiarities, results and perspectives." Signal Processing: Image Communication 30 (2015): 57-77.

[14] N. Ponomarenko, O. Ieremeiev, V. Lukin, K. Egiazarian, L. Jin, J. Astola, B. Vozel, K. Chehdi, M. Carli, F. Battisti, C.-C. Jay Kuo, Color Image Database TID2013: Peculiarities and Preliminary Results, Proceedings of 4th Europian Workshop on Visual Information Processing EUVIP2013, Paris, France, June 10-12, 2013, pp. 106-111.

[15] N. Ponomarenko, O. Ieremeiev, V. Lukin, K. Egiazarian, L. Jin, J. Astola, B. Vozel, K. Chehdi, M. Carli, F. Battisti, C.-C. Jay Kuo, A New Color Image Database TID2013: Innovations and Results, Proceedings of ACIVS, Poznan, Poland, Oct. 2013, pp. 402-413.

7. APPENDEX: CODE

- The Code folder contains 4 folders
  1. Part 1
  2. Tid2013
  3. SSIM
  4. Edge_detection

Part 1 contains the following files inside src folder

__init__.py	//main file to perform all the image filtering operation
Convolution.py	//file contains Gaussian Filter implementation
Binarization.py	//file contains Otsu implementation
ColoringComponents.py	//file containing the two-pass approach for finding //connected component

Tid2013

This folder contains the images from TID2013 database and other relevant information related to TID

SSIM

__init__.py	//file contains the implementation of SSIM, SSIM-//WINDOW SPLIT and RMS
Binarization.py	//file contains Otsu implementation

Edge_detection

sobel_edge.py	//file contain the implementation of Sobel edge //detection algorithm