

# Gaussian based Image Segmentation Algorithm

Adolf Fenyi

Sefwi Wiawso College of  
Education

P.O Box 94, Sefwi Wiawso, Ghana

Emmanuel Nkansah

Sefwi Wiawso College of  
Education

P.O Box 94, Sefwi Wiawso, Ghana

Ebenezer Eghan

Sefwi Wiawso College of  
Education

P.O Box 94, Sefwi Wiawso, Ghana

## ABSTRACT

This study adopted the Gaussian equation to eliminate noisy pixels in a segmented image. The equation was converted into a convolution kernel and the values were normalized to attain a perfect result. In order to preserve every information in the image, the border pixels were padded with zeros before the image was convolved with the designed kernel. A variance was computed from the result obtained from the convolution operation and the pixel that obtained the minimum variance was selected as the threshold. In order to obtain the optimum threshold, the computed threshold was multiplied by the optimization constant which ranges from 0.1 to 1, and the resultant was considered as the final threshold for the segmentation process.

The Gaussian algorithm was evaluated with global, Otsu and adaptive algorithms. The performance metrics used were signal to noise ratio (defines the sensitivity of algorithm), mean square error and running time (specifies the execution time of an algorithm). In the first experiment which was made up of 20 noise free images, the adaptive attained the highest rating of 7.117dB. This was followed by the Gaussian algorithm at 5.231dB. The poorest performance was seen in the global at 1.988dB. The second experiment was done with ten noisy images. In the experiment, the Gaussian recorded the highest rating of 3.124dB, while the adaptive scored the poorest at 1.847dB.

From the running time, the fastest algorithm was seen in the global at 2.814s. This was followed by the Otsu at 9.814s. There was only a difference of 2.107s between the average execution time of the Otsu and the Gaussian. This was due to the noise suppressing mechanism in the Gaussian. The slowest algorithm was the adaptive which recorded a running time of 110.594s

The Gaussian algorithm performed better when there was presence of noise in the image. This explains why it recorded sensitivity rate of 3.124dB in the second experiment. However, for noise free images, the adaptive algorithm is the best despite the poor performance in the running time.

## General Terms

Image processing, Algorithms, Segmentation, Gaussian

## Keywords

Global, Adaptive, Otsu, PSNR, SNR, Segmentation, Running Time

## 1. INTRODUCTION

Binary image analysis reduces several sections that are found in an image into a small form that will show in the segmented image. The main aim of performing image segmentation is to ensure that the object of interest is isolated from its background into a relevant form that can be understood easily. Even though object recognition is considered as the essential

step in image segmentation, it should be known that it is one of the challenges affecting computer vision. Each section of an image is made up of a number of pixels which contains important information about the object in the image[1]. One of the techniques applied to image segmentation is thresholding, because it is easy and efficient to implement. Edge detection is another important technique in image segmentation. The challenge encountered in conducting a research based on edge detection is the presence of edges in areas where borders does not exist[2].

## 1.1 Importance of Image Segmentation

Image segmentation is used in a lot of fields which include medical imaging, traffic control system, object detection and video surveillance.

### 1.1.1 Medical Imaging

Medical imaging devices are controlled by a number of segmentation algorithms. Applications such as Magnetic Resonance Imaging and Computer Tomography generate images which are hard to scrutinize due to the high image resolutions. Segmentation algorithms assist clinicians to visualize complex images by extracting important regions in the image. This region of interest could be tumors or other models of organs. This procedure helps medical staffs to perform quantification and detailed analysis of a particular case.

### 1.1.2 Vehicular Identification in the Presence of Occlusion

Classification of vehicles, speed and count are very important for transportational projects which include traffic light and road constructions. Vehicle identification devices use ultrasonic sensors, radars, and lasers to operate. This technology combines the driver's behaviour pattern to the colour information of the vehicles. In order to find a moving vehicle, a camera is positioned in a car to face the object of interest and the identification process is initiated by Support Vector Machines. Vehicles which are in traffic are captured by cameras positioned near the roads. These images are later run through segmentation algorithms for extracting important features for decision making.

### 1.1.3 Identification of Object in Satellite Images

Water bodies, soil types, vegetation and other areas are known through the help of satellite imaging. The segmentation algorithms which are applied are clustering based and histogram techniques. The clustering based method[3] converts image pixels into points. These points are arranged into clusters using their colour information. The distribution of points in the clusters is influenced by the noise level and shading effects of the acquisition device. A typical example of the clustering based method is the K-means algorithm which is applicable in vectorization, image segmentation and data compression.

The histogram based method[4] is also used in identifying objects in satellite images. The method maps the pixels in the image onto an histogram. This helps the objects pixels to be separated from its background on the image. For 3D images, the histogram generates a lot of spurious peaks which result in ambiguous segmentation.

## 1.2 Problem Statements

Poor histogram distribution, non-uniform illumination, noise and high time complexity are some of the challenges that affect image segmentation.

### 1.2.1 Non-uniform illumination

Illumination changes across a scene is a big problem in binary image segmentation. This problem often causes parts of the object in the light regions to be brighter than others[5]. Since the global thresholding could not handle this problem, the adaptive was built[6]. This algorithm adopts the chow and Kaneko [7] techniques to calculate the threshold value. The purpose of this algorithm is that minute regions in an image are likely to have uniform illumination. This algorithm divides an image into cluster of sub images. It then computes the optimum threshold for each sub image for the purpose of the segmentation process. This algorithm is not good for real time applications since it has a high time complexity.

### 1.2.2 Noise

One of the problems of image segmentation is noise. It is introduced by capturing devices which include scanners and digital cameras. It is also known as image by product because of the spurious information it adds to images[8]. An image noise could be specks which are presence on images captured under good conditions to astronomical images where little information could be obtained from complex processing. Some of the examples of noise that affect image quality are Gaussian, uniform, shot and speckle noise.

### 1.2.3 Poor Histogram distribution

This problem is found in histogram based methods[9]. These methods use valleys and peaks which are obtained from the histogram to compute the threshold. The segmentation process becomes simplified when only two classes of pixels are generated from the histogram. The problem with Otsu algorithm is that poor results are generated when a lot of classes are obtained from the histogram.

### 1.2.4 Time Complexity

This defines the amount of time a segmentation algorithm is required to execute[10]. When this is expressed in big O notation it removes lower order terms and coefficients. The time complexity of an algorithm with input size  $4m^2+2m$  is  $O(m^2)$ . The time complexity of an algorithm is determined by the total number of elementary operations. The 'tic' and 'toc' functions which are found in the Matlab application is used to determine the running time of an algorithm in the Matlab. Most of the segmentation algorithms consume valuable resources from the machine they run on due to their high time complexity. These resources include memory and CPU. Atypical example is the iterative thresholding which only stops execution when there is insignificant difference between two successive thresholds. The algorithm also takes a long time to compute the threshold at each execution since the image is not partitioned into small blocks of pixels.

## 1.3 The Objectives

The aim of this research is to attain the following:

- Eliminate the effect of poor illumination in image segmentation
- Reduce the time complexity of segmentation algorithms
- Reduce the effect of noise in image segmentation

## 2. LITERATURE REVIEW

This section will review some of the common segmentation algorithms which are global, adaptive, Otsu and iterative.

### 2.1 Global

The simplest thresholding technique is the global where a fixed value is used to partition the image into two classes. A captured image is scanned and each pixel is labeled either as a background or object. For this algorithm to work properly, the image should have two dominant modes[11]. This algorithm is successful in industrial inspection applications. Global thresholding suffers from non-uniform illumination which causes portions of the image to be brighter than others. These darker and brighter parts may not relate to the actual object in the image.

### 2.2 Iterative Method

Iterative algorithm works best on images with bimodal histogram[12]. This algorithm will continue to execute until an insignificant threshold is discovered between two successive thresholds. It takes a long time to compute a threshold at each phase of the algorithm due to the fact that the image is not partitioned into small blocks of pixels. It however increases its time complexity, thereby consuming more memory from the computer. It is good for applications that require a high computational time.

### 2.3 Adaptive Method

This algorithm was proposed by Rosenfeld and Nakagawa[13]. These two researchers partitioned the image into a number of windows based on their locality properties. A threshold is then estimated for windows having bimodal histograms. A final threshold is calculated for the entire image by interpolating the thresholds from the various windows. The adaptive technique is able to solve the problems pertaining to non-uniform illumination in binary images[14].

### 2.4 Multiple Method

This technique is used to segment a binary image into a number of distinct areas. This algorithm is very good for objects having colored or complicated backgrounds, since the global algorithm is unable to handle such task.

A multiple threshold image  $h(m,n)$  is represented by the function:

$$h(m,n) = \begin{cases} x; g(m,n) < T4 \\ y; T3 < (m,n) \leq T4 \\ z; g(m,n) \leq T3 \end{cases}$$

Multilevel thresholding groups the coordinate  $(m,n)$  such that if  $(m,n)$  is greater than  $T3$  but less than  $T4$ , then it belongs to one object group, and vice versa.

### 2.5 Otsu Method

This algorithm was developed by Nobuyuki Otsu to overcome the challenges with clustering based thresholding. In order for this to work properly, the image should consist of only two classes namely, background and foreground pixels and the histogram of the image should also be bi-modal. The algorithm then calculate the optimum threshold by

partitioning the foreground and background classes such that they have a minimum intra class variance[15]. This is the most widely used and accepted algorithm due to its effectiveness and simplicity[16]. This algorithm fails when there is a big difference between an object pixels and its background. In terms of selecting a suitable threshold, the algorithm is considered as the best choice. This algorithm spends a long time to execute multi-level images due to number of classes involved.

The algorithm uses the steps below for its execution

- Calculate the probability of the intensity levels
- Select the initial mean and the probability
- Update the initial mean and the probability by iterating through the thresholds
- Calculate the threshold's variance
- Select the threshold having the maximum variance as the final threshold.

The challenge with Otsu algorithm is that it does not work properly on images having more than two classes. However, in the real world, most of the images contain more than two classes of pixels.

### 3. METHODOLOGY

This section uses the Gaussian algorithm to mitigate the challenges in image segmentation. Some sets of images and the Matlab application were used for the implementation[17].

The implementation steps are outlined below:

- Convert the Gaussian equation into a convolution kernel. The equation is represented below:

$$G(x, y) = \left( \frac{1}{2\pi\delta^2} \right) e^{-\frac{(x^2+y^2)^1}{2\delta^2}} \quad (3.1)$$

Where (x,y) signifies the pixels index, e represents Euler's constant, which is 2.718, and  $\delta$  represents the standard deviation which is 2.4.

- Normalize the kernel values to attain a total weight of 1
- Pad the image with enough pixels so that border pixels will have the required neighbours to form a matrix of 3 X 3. This operation is necessary since it will prevent any information lost during the segmentation process.
- Convolve the matrix pixel values with the kernel designed from equation 3.1, and assigned the result to the center pixels. The convolution operation is expressed in equation 3.2

$$P(x, y) = \sum_{x=-2, y=-2}^{x=2, y=2} P * T(x, y) \quad (3.2)$$

Where T signifies the convolution kernel and (x,y) represents the image 'P' index.

- Convert the result generated from the convolution operation into integers and calculate the variance.

The variance is calculated from equation 3.3

$$\sigma = \sum_{c=q}^D \frac{(c - u)^2}{W} \quad (3.3)$$

Where c represents intensity values of pixels starting from the least (q) to the highest (D),  $\sigma$  is the variance, W is the frequency, and u signifies the mean of the pixels.

- Select intensity value that yields the minimum variance as the threshold for the segmentation process.
- Compute the optimum threshold from equation 3.4

$$L_o = L * O_p \quad (3.4)$$

Where  $L_o$  is the final threshold, L is the threshold that generated the minimum variance and  $O_p$  is the optimization constant between 0.1 to 1.

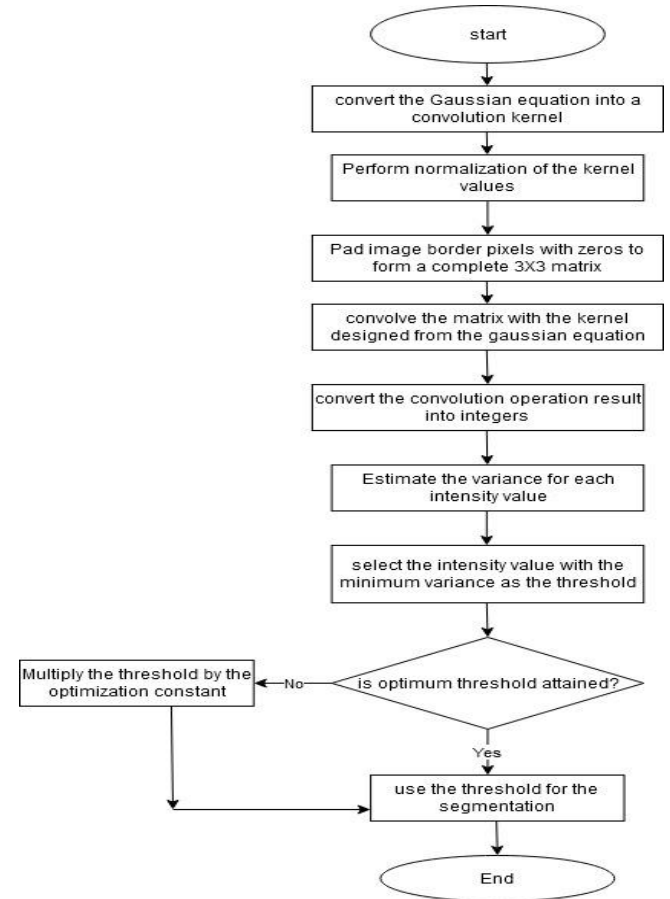


Fig 1: The flowchart of the Gaussian Algorithm

### 4. RESULT ANALYSIS AND DISCUSSIONS

This section will evaluate the Gaussian algorithm with the adaptive, global, and the Otsu to determine the best one. The materials for the evaluation are Matlab program, segmentation algorithms implementation and some images from Berkeley Segmentation Dataset. Signal to Noise Ratio and execution time were used as the performance metrics for the evaluation. The Signal to Noise Ratio is represented by equation 4.1

$$SNR = 10\log\left(\frac{SP}{NP}\right) \quad (4.1)$$

Where SP represents signal power which is the mean of the intensity values while NP denotes noise power, which is standard deviation of the intensity values.

The tic and toc functions in the Matlab application enabled the execution time to be determined. The timer is initiated by the tic command while the toc reads the elapsed time.

The total images used for the evaluation were thirty two (32). Twenty of these images were noise free, while the remaining were polluted with uniform, poisson and gaussian noise. The reason for this pollution was to verify the performance of the algorithms in the presence of noise. The evaluator computes SNR, MSE (Mean Square Error) and running time of a segmented image from an algorithm. It then plots a graph with the vertical axis representing the performance metrics. The study computed an average performance for all the various image categories to determine the true performance of the algorithms, since only one evaluation cannot be done to conclude on the true performance of the algorithms.



Fig 2: The image of a canoe and a building

Table 1: Assessing the segmentation algorithms performance for a noise free image

Algorithms	Signal to noise ratio (SNR)	Mean Square Error (MSE)
Otsu	3.456	0.647
Global	2.141	0.772
Adaptive	6.417	0.243
Gaussian	4.517	0.347

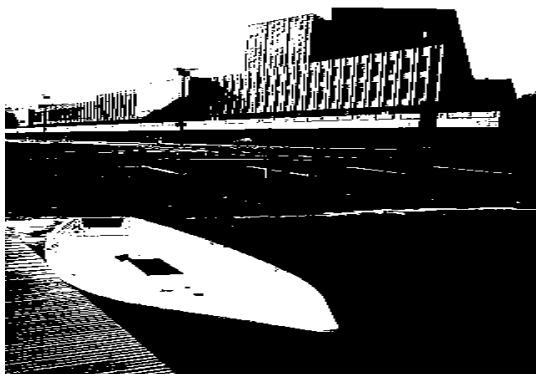


Fig 3: The output of Otsu

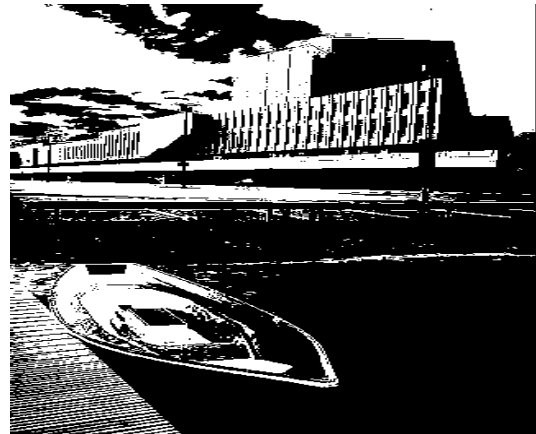


Fig 4: The output of the Global

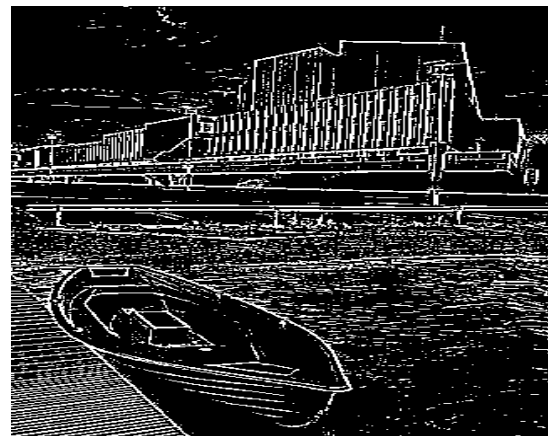


Fig 5: The output of the Adaptive

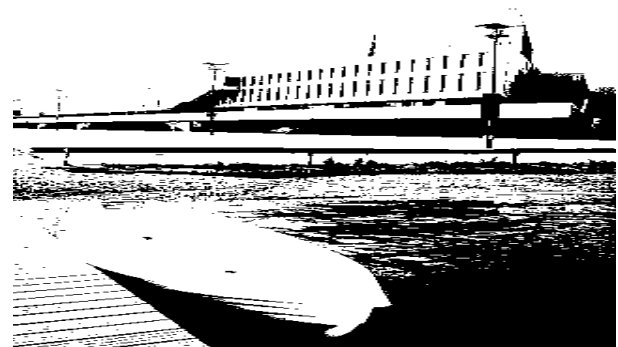


Fig 5: The output of the Gaussian

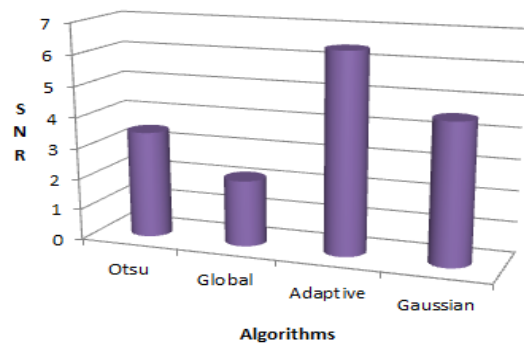


Fig 6: Graphical representation of the algorithms performance



Fig 7: An image of a horse polluted with the uniform noise

Table 2: Assessing the segmentation algorithms performance for a uniform polluted image

Algorithms	Signal to noise ratio	Mean square error
Otsu	1.795	0.521
Global	0.832	0.645
Adaptive	1.927	0.477
Gaussian	2.762	0.384

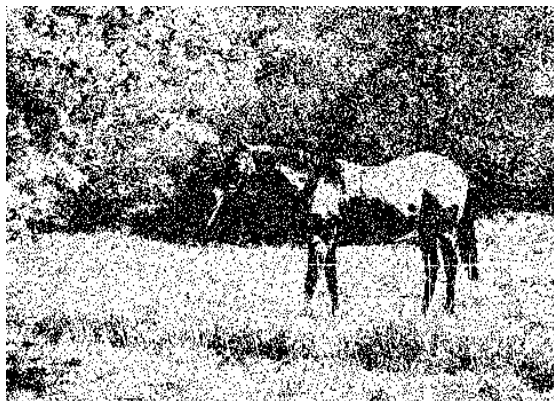


Fig 8: The output of the Otsu



Fig 9: The output of the Global

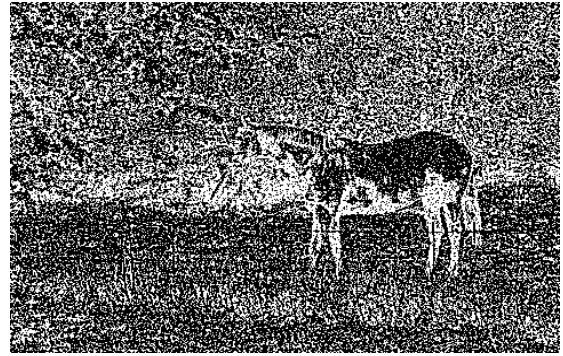


Fig 9: The output of the Adaptive



Fig 10: The output of the Gaussian

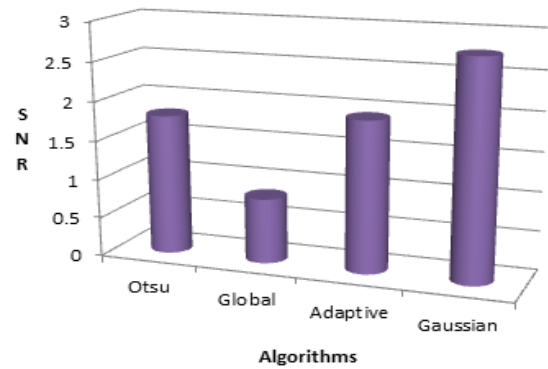


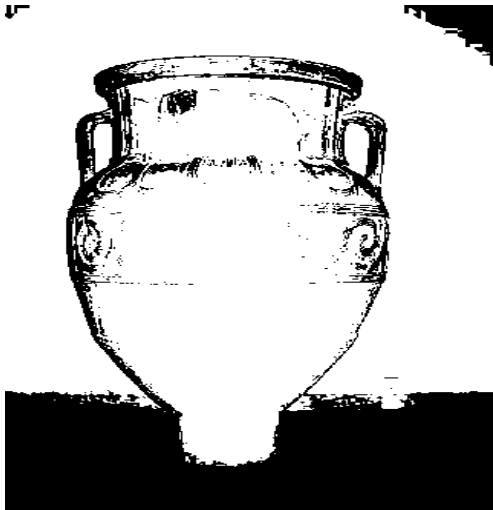
Fig 11: Graphical representation of the algorithms performance



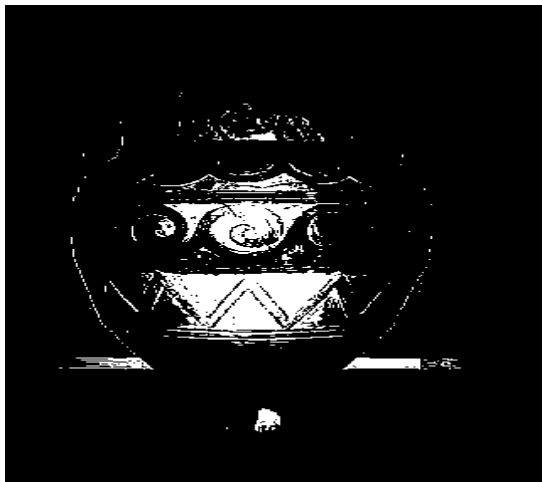
Fig 12: An image of a pot

**Table 3: Assessing the execution time of the segmentation algorithms**

Algorithms	Execution Time 1	Execution Time 2	Execution Time 3	Average Execution Time
Otsu	10.527	9.847	10.117	10.163
Global	3.471	3.127	2.981	3.193
Adaptive	110.70	100.981	99.241	103.640
Gaussian	12.417	13.142	12.984	12.848



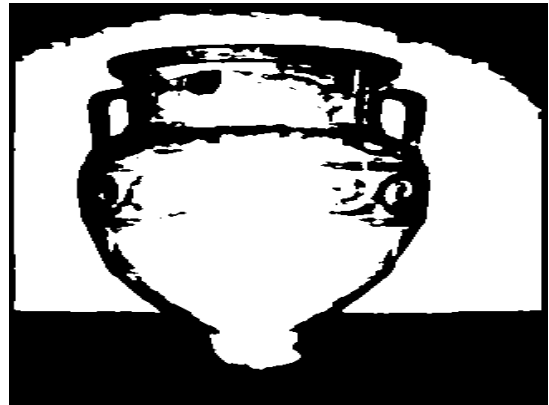
**Fig 13: The output of the Otsu**



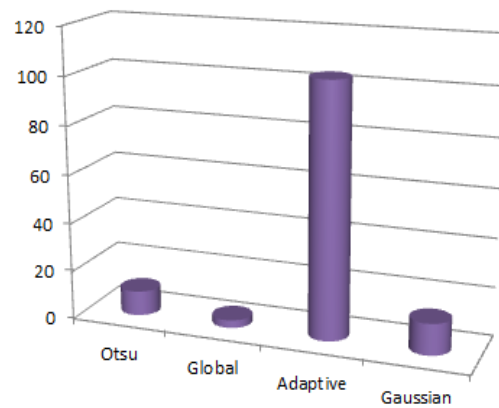
**Fig 14: The output of the Global**



**Fig 14: The output of the Adaptive**



**Fig 15: The output of the Gaussian**



**Fig 16: Graphical representation of the algorithms execution time**

## 5. CONCLUSION

The technique that enables an image to be partitioned into many related portions is image segmentation. This research studied the Adaptive, Otsu, global and the Gaussian algorithms. Image segmentation is applicable in a lot of areas which include facial detection, medical imaging, identification of objects in satellite imaging and vehicle identification in the presence of occlusion.

Noise, non-uniform illumination, poor histogram distribution and high time complexity are some of the challenges that affect image segmentation. This study adopted the Gaussian equation to mitigate the effect of noisy image pixels.

The equation was converted into a convolution kernel and the values were normalized to attain a perfect result. In order to preserve every information in the image, the border pixels were padded with zeros before the image was convolved with the designed kernel. A variance was computed from the result obtained from the convolution operation and the pixel that obtained the minimum variance was selected as the threshold.

In order to obtain the optimum threshold, the computed threshold was multiplied by the optimization constant which ranges from 0.1 to 1, and the resultant was considered as the final threshold for the segmentation process.

In the Result Analysis and Discussion section, the Gaussian based algorithm was compared with the Global, Otsu, and Adaptive. The performance metrics used were signal to noise ratio, mean square error and execution time. In the first experiment that consisted of twenty noise free images, the Adaptive attained the signal to noise ratio of 7.117dB. This was followed by the Gaussian algorithm at 5.231dB. The poorest performance was seen in the global at 1.988dB.

The second experiment was done with ten noisy images which were polluted with uniform, poisson and Gaussian noise. In the experiment, the Gaussian based algorithm recorded the highest rating of 3.124dB, while the adaptive scored the poorest at 1.847dB. When the average performance of the second experiment is compared with the first, it could be seen that the first experiment was ahead by 8.162dB. This shows how poorly all the algorithms performs in the presence of noise.

The third experiment estimated the running time of the various algorithms. From the attained result, the poorest algorithm was the adaptive which was 110.594s. This is because the algorithm divides a selected image into several regions, and estimate thresholds for the various segments which are later used for the segmentation process. This operation does not only increase the time complexity but also consumes so many resources (memory and CPU) from the computing device. This was followed by the Gaussian based algorithm at 11.921s. It uses the Gaussian equation to suppress noise which could affect the visibility of segmented image. Therefore, it performs better when there is the presence of noise in the image. This explains why it recorded sensitivity rate of 3.124dB in the second experiment. However, for noise free images, the adaptive algorithm is the best despite the poor performance in the running time.

The global was seen as the fastest algorithm with an average score of 2.814s. This is because the algorithm computes only a single threshold for segmenting the entire image which in most cases leads to a poor result. Since this is non-difficult task, a small overhead is experienced by the executing computer.

## 6. REFERENCES

- [1] Acharya, T., & Ray, A. K. (2005). *Image Processing: Principles and Applications* (Vol. 15). <https://doi.org/10.1117/1.2348895>
- [2] Dass, R., & Devi, S. (2012). Image Segmentation Techniques. *International Journal of Electronics & Communication Technology (IJECT)*, 3(1), 66–70.
- [3] Barghout, L., & Sheynin, J. (2013). Real-world scene perception and perceptual organization: Lessons from Computer Vision. *Journal of Vision*, 13(9), 700–710.
- [4] ] Shapiro, L. G., & Stockman, C. G. (2001). *Computer Vision*. New Jersey: Prentice-Hall Publisher.
- [5] Jain, A. (1986). *Fundamentals of Digital Image Processing*. New Jersey: Prentice-Hall Publisher.
- [6] Davies, E. (1990). *Machine Vision: Theory, Algorithms and Practicalities*. Cambridge, Massachusetts: Academic Press Publisher.
- [7] Chow, C.K., & Kaneko, T. (1972). Automatic Boundary Detection of the Left Ventricle from Cineangiograms. *Computers and Biomedical Research*, 4(9), 303–356.
- [8] Rohankar, J. (2013). Survey of the Various Noises and Techniques for Denoising the Colour Image. *International Journal of Application or Innovation in Engineering & Managemen*, 2(11), 257–289.
- [9] Shapiro, L. G., & Stockman, C. G. (2001). *Computer Vision*. New Jersey: Prentice-Hall Publisher.
- [10] Sipser, M. (2006). *Introduction to the Theory of Computation*. Boston MA, USA: Course Technology Publisher.
- [11] Rafael C. Gonzalez, R. E. W. (2001). *Digital Image Processing* (3rd ed.). Upper Saddle River, New Jersey 07458: Prentice Hall. Retrieved from <http://www-cs.cny.cuny.edu/~zhu/Capstone2007/DIP-Chapter10-Segmentation-notes.pdf>
- [12] Bansal, S., & Maini, R. (2013). A Comparative Analysis of Iterative and Ostu ' s Thresholding Techniques. *International Journal of Computer Applications*, 66(12), 45–47.
- [13] Rosenfeld, Y. N. and A. (1979). Some experiments on variable thresholding. *Pattern Recognition*, 11(3):191–204.
- [14] Rafael C. Gonzalez, R. E. W. (2001). *Digital Image Processing* (3rd ed.). Upper Saddle River, New Jersey 07458: Prentice Hall. Retrieved from <http://www-cs.cny.cuny.edu/~zhu/Capstone2007/DIP-Chapter10-Segmentation-notes.pdf>
- [15] Sunil L. Bangare, Amruta Dubal, Pallavi S. Bangare, D. S. T. P. (2015). Reviewing otsu ' s method for image thresholding, (August 2016).
- [16] Vala, M. H. J., & Baxi, A. (2013). A review on Otsu image segmentation algorithm. *International Journal of Advanced Research in Computer Engineering and Technology*, 2(2), 387–389.
- [17] Martin, D., Fowlkes, C., Tal, D., & Malik, J. (2001). A Database of Human Segmented Images and its Application to Evaluate Segmentation Algorithms and Measuring Ecological Statistics. *Proceedings of 8th International Conference on Computer Vision* (pp. 416--423).