Tribhuvan University

Institute of Science and Technology

A Comparative Analysis of Otsu Thresholding and K-means Algorithm for Image Segmentation

Dissertation

Submitted to

Central Department of Computer Science and Information Technology

Kirtipur, Kathmandu, Nepal

In Partial Fulfillment of the Requirements

For the Master's Degree in Computer Science and Information Technology

By

Kshitiz Bhatt

T.U Registration No: 5-2-61-60-2011

T.U. Examination Roll No: 314/073

December, 2019

Supervisor

Mr. Bikash Balami

Tribhuvan University

Institute of Science and Technology

Central Department of Computer Science and Information Technology

Student's Declaration

I hereby declare that I am the only author of this work and no sources other than listed here have been used in this work.

Kshitiz Bhatt

Date: December, 2019

Tribhuvan University

Institute of Science and Technology

Central Department of Computer Science and Information Technology

Supervisor's Recommendation

I hereby recommend that this dissertation is prepared under my supervision by **Mr. Kshitiz Bhatt** entitled **"A Comparative Analysis of Otsu Thresholding and K-means Algorithm for Image Segmentation"** be accepted as in fulfilling partial requirement for the completion of Master's Degree of Science in Computer Science and Information Technology.

____

Asst. Prof. Bikash Balami

Central Department of Computer Science and Information Technology

Tribhuvan University, Kathmandu Nepal

(Supervisor)

Date: December, 2019

Tribhuvan University

Institute of Science and Technology

Central Department of Computer Science and Information Technology

LETTER OF APPROVAL

We certify that, we have read this dissertation and in our opinion it is satisfactory in the scope and quality as a dissertation in partial fulfillment for the requirement of Master's Degree in Computer Science and Information Technology.

Evaluation Committee

____

Asst. Prof. Bikash Balami
Central Department of Computer Science &
Information Technology
Tribhuvan University
(Supervisor)

____

Asst. Prof. Nawaraj Paudel
Head of Department
Central Department of Computer Science &
Information Technology
Tribhuvan University

____

External Examiner

____

Internal Examiner

ACKNOWLEDGEMENT

I would like to offer gratitude to my supervisor Asst. Prof. Bikash Balami Central Department of Computer Science and Information Technology, Tribhuvan University for his persistence help, clear guidance and encouragement throughout my research for this thesis. His suggestions and guidance at every stage is one of the major reasons of the successful completion of my thesis.

I would also like to thanks respected Head of Department Asst. Prof. Nawaraj Paudel and faculties for their guidance and help in every aspect of my academic study and providing me the opportunity to pursue my Master's degree in such wonderful academic environment.

I would like to express my sincere thanks to my colleagues Mr. Bhoj Raj Adhikari, Bikash Regmi, Sushil Banstola, and all my well-wishers who directly and indirectly helped me to complete this work.

Last but not least I must express my very profound gratitude to my parents and Sister Kritika Bhatt for providing me with unfailing support and continuous encouragement through the process of writing this thesis.

Abstract

Image segmentation is an aspect of image processing and is used to find out the object in the image and dividing the image into different segments and discrete regions. The goal of image segmentation is to change the representation of an image into something that is more meaningful and easier to analyze and usually serves as the pre-processing before pattern recognition, feature extraction, and compression of the image. There are several challenges emerged in the field of image segmentation such as to differentiate between regions that may be determined by different factors, such as color, gray level or texture, overlapping objects can be difficult to separate also, shadowing can create additional borders. Many kinds of research have been done in the area of image segmentation. This research evaluates the two image segmentation algorithms Otsu thresholding and K-means using a parameter: Peak Signal to Noise Ratio (PSNR), Mean Square Error (MSE), Structural Content (SC) and Structural Similarity Index (SSIM) to measure the quality of segmented image. The lower value of MSE shows that the higher quality of segmented image is obtained. The higher value of PSNR and SSIM shows that for K-means original and segmented images do not loss the much of the property and have the higher similarities in input and segmented image. The lower SC value of an input and segmented image indicate the better quality of the image is generated by Otsu thresholding.

Keywords:* *Image Segmentation, MSE, PSNR, SSIM, SC, K-means, Otsu thresholding,

Table of Contents

List of Tables

List of Figures

List of Abbreviations

GB	Giga Bytes
HSV	Hue, Saturation, Value
IDE	Integrated Development Environment
MSE	Mean Square Error
PSNR	Peak Signal to Noise Ratio
RAM	Random Access Memory
RGB	Red, Green, Blue
SC	Structural Content
SSIM	Structural Similarity Index

Chapter 1
Introduction

1.1 Introduction

Image segmentation is an approach for the decomposition of the digital image into various different sets of pixels, generally called segments. The pixels in a region are similar in accordance to some homogeneity standards such as color, intensity or texture, so as to locate and identify objects and boundaries in an image [16]. It is considered as an important operation for meaningful analysis and interpretation of the image. It is a critical and essential component of an image analysis or pattern recognition system, and is one of the most difficult and an important task in image processing.

The principal purpose of the segmentation approach in image processing is to make the image representation so easy for the analysis. It is the prime area of research in computer vision and is defined as the process of partitioning an image into meaningful regions or objects. The various fields of applications for segmentation includes locating tumors, measure tissue volume, computer-guided surgery, study of anatomical structure, locate objects in satellite images and fingerprint recognition, face detection etc. [12].

Segmentation methods are categorized on the basis of discontinuity and similarity property. The methods that are based on a discontinuity property of pixels include edge, line, and point detection techniques. The edge-based method attempts to resolve image segmentation by detecting the edges or pixels between different regions that have rapid transition in intensity and are extracted and linked to form closed object boundaries [2, 9].

Discontinuity approach is to partition an image based on sudden changes in gray scale. The principal areas of interest include detection of isolated point's detection of lines and edges in image. There are three basic types of grey level discontinuities in digital images which includes points, lines and edges. In similarity detection-based approach the digital image is segmented in different regions on the basis of similarities like utilizing a similar set of pixels. Under this approach various methods like thresholding techniques, clustering techniques, region merging, region splitting and region growing techniques are categorized [5].

1.1.1 Thresholding

One of the most popular approaches for image segmentation is through thresholding. The basic idea of thresholding is quite simple. It involves selecting an optimal gray-level threshold value for separating interested object in an image from the background based on their gray-level distribution. Thresholding takes a gray-scale image and replaces each pixel with a black one if its intensity is less than some fixed constant, or a white pixel if the intensity is greater than that constant. The new binary image produced separates dark from bright regions. Mainly because finding pixels that share intensity in a region is not computationally expensive, thresholding is a simple and efficient method for image segmentation [9].

Formally, a function g(x; y) can be defined over every pixel value of the original image f(x; y) that defines the new thresholding image.

$$g(x,y) = \begin{cases} 1 \ \ if\ f(x,y) \geq T \\ 0\ if\ f(x,y) < T \end{cases} \qquad \text{eq.(1)}$$

Equation (1) defines that for every pixel on the original image, a new value of 0 or 1 will be assigned to the new image, depending if the current pixel value is greater than some defined threshold T.

1.1.2 Otsu Thresholding

Otsu's algorithm is a simple and popular thresholding method for image segmentation. The algorithm divides the image histogram into two classes; by using a threshold such as the intra-class variance is very small i.e. as compact as possible. Otsu's algorithm does not take account on the spatial relationship between pixels of image. So regions that have similar pixel values, but are in completely different locations in the image are merged when computing the histogram.

Otsu method is a type of global thresholding that depends on gray value of the image. The Otsu thresholding is depends on gray value of the image. The initial value for the threshold is 1 and algorithm checks the between class's variance and within class's for every value of t from 1 to the largest pixel value. The algorithm presumes that the image have bimodal histogram and further evaluates the optimum threshold, partitioning two classes (background and foreground of an image) so that their intra-class variance is negligible. Otsu's method minutely explores the threshold that minimizes the intra-class variance, defined as a weighted sum of variances of the two classes:

$$\sigma_\omega^2\ (t) = \ \omega_1(t)\sigma_1^2(t)\ +\ \omega_2(t)\sigma_2^2(t) \qquad \text{eq.(2)}$$

Where σ_ω^2 is the within class variance, weights ω_i are the probabilities of the two classes separated by a threshold t and σ_i^2 are variances of these two classes.

A probability function P is obtained for every pixel value. First, the histogram distribution for the image is computed, and then, normalization is performed in order to guarantee it follows a probability distribution. After that, the pixel values are divided into two classes C1 and C2 by a threshold t, using the class probability functions $\omega_1(t)$ and $\omega_2(t)$as:

$$\omega_1(t) = \sum_{i=1}^{t} P(i) \qquad \text{eq.(3)}$$

$$\omega_2(t) = \sum_{i=t+1}^{I} P(i) \qquad \text{eq.(4)}$$

Class C1 represents those pixels with intensity levels in [1; t], and class C2 represents those pixels with levels in the interval [t + 1; I], where I is the largest pixel value. Then, the means for class C1, $\mu_1(t)$, and class C2, $\mu_2(t)$ are obtained:

$$\mu_1(t) = \sum_{i=1}^{t} iP(i) / \omega_1(t) \qquad \text{eq.(5)}$$

$$\mu_2(t) = \sum_{i=t+1}^{I} iP(i) / \omega_2(t) \qquad \text{eq.(6)}$$

After that, the variance for class C1, $\sigma_1^2(t)$, and class C2, $\sigma_2^2(t)$ are computed

$$\sigma_1^2(t) = \sum_{i=t}^{I} [i - \mu_1(t)]^2 P(i)/\omega_1(t) \qquad \text{eq.(7)}$$

$$\sigma_2^2(t) = \sum_{i=t+1}^{I} [i - \mu_2(t)]^2 (i)/\omega_2(t) \qquad \text{eq.(8)}$$

Otsu displays that by minimizing the intra-class variances is same as maximizing the inter-class variance ($\sigma_b^2(t)$) as:

$$\sigma_b^2(t) = \omega_1(t)\omega_2(t)[\mu_1(t) - \mu_2(t)]^2 \qquad \text{eq.(9)}$$

1.1.3 K-means Clustering

Clustering is a method to divide a set of data into a specific number of groups and one of the popular methods to do so is k-means clustering. K-means is an iterative algorithm in which it minimizes the sum of distances from each object to its cluster centroid, over all clusters. The k-means clustering, partitions a collection of data into a k number group of data. It classifies a given set of data into k number of disjoint clusters. K-means algorithm consists of two separate phases. In the first phase, k-means algorithm calculates the k centroid and in the second phase it takes each point to the cluster, which has nearest centroid from the respective data point iteratively. Based on the centroid value calculated, a new distance is calculated between each center and each data point and assigns the points in the cluster which have minimum distance [8]. In K-means method the distance of nearest centroid is generally calculated using Euclidean distance.

1.2 Problem Statement

The problem is to pick some relevant objects from the background in an image. Image segmentation allows marking important objects or regions for further analysis. The problem is to pick some relevant objects from the image. The image analysis algorithm should find the edge of objects in images like bioinformatics as tumor mass in an X-ray image, a tooth root canal or a component of a printed circuit board properly and separate the regions without adding or subtracting any extra information. Incorrect segmentation in these cases may result in inadequate treatment or in the mislabeling of vital components.

1.3 Objective

The objective of the research is

- To analyze the image segmentation using Otsu thresholding and k-means algorithm techniques.
- To evaluate the performance of the algorithms.

1.4 Thesis Organization

The organization of this thesis is as follows:

Chapter 1 describes the introduction, problem statement and objectives.

Chapter 2 describes the literature review of the existing researchers related to image segmentation.

Chapter 3 describes the algorithm and methodology of the Otsu thresholding and K means clustering algorithm.

Chapter 4 contains the implementation overview of the algorithms with result analysis.

Chapter 5 concludes the conclusion of thesis works.

Chapter 2

Background Study and Literature Review

2.1 Background

Image segmentation is one of the fundamental techniques used to subdivide an image into its integrant portions to extract relevant information from the image. Though in real-world case, image segmentation performed effective roles in several demanding applications such as image retrieval, automatic traffic control, medical imaging, object detection, and video surveillance etc, image segmentation is very challenging due to the corruption by artifacts such as partial volume effect, image noises and bias field effects, image degradation such as blurring, contrast or color imperfection, image texture or structure variability and complexity etc. The problem of image segmentation has been known and addressed for the last many years. But, it still considered being one of the most difficult and challenging tasks in image processing and object recognition, and determines the quality of final results of the image analysis.

2.2 Literature Review

There have been many works done in the area of image segmentation by using different methods. And many are based on different application of image segmentation. K-means algorithm is the one of the simplest clustering algorithm and there are many methods implemented so far with different methods to initialize the center. And many researchers are also trying to produce new methods which are more efficient than the existing methods. K-Means algorithm is an unsupervised clustering algorithm that classifies the input data points into multiple classes based on their inherent distance from each other. The iterative k-means clustering algorithm was first proposed by MacQueen (1997).

The paper [10], introduced a new, efficient approach towards K-means clustering algorithm. A new method for generating the cluster center by reducing the mean square error of the final cluster without large increment in the execution time has been proposed. It reduced the mean square error without sacrificing the execution time. Many comparisons have been done and it can conclude that accuracy is more for dense dataset rather than sparse dataset.

The paper [18], proposed enhancing K-means clustering algorithm with an improved initial center. A new method for finding the initial centroid is introduced and it provides an effective way of assigning the data points to suitable clusters with reduced time complexity. In the study algorithm has been proposed that has more accuracy with less computational time comparatively with original k-means clustering algorithm. This algorithm does not require any additional input like threshold value. But this algorithm still initializes the number of cluster k and suggested determination of the value of k as one of the future work.

A class variance method known as the Otsu's method was presented by Nobuyuki Otsu in the year 1979, for the ease of calculations, consistency and effectiveness. The Otsu's method is one of the effective processes employed for the selection of threshold and is well known for its rare time consumption [6].

The paper [11], proposed 1-Dimensional Otsu algorithm which is being widely used because it's simple, efficient and reliable in nature. 1-Dimensional Otsu algorithm merely examines the gray level information of the image pixel disregarding spatial neighborhood information which can cause low quality segmentation. According to the Zhong Qu and Li Zhang the main function of Otsu method is like as the K-means method of multilevel thresholding. Both the approaches are focused on the within class variance minimization. The Otsu method of image segmentation performs the segmentation of digital image using gray level histogram while this is not so in a K-means method of image segmentation.

A gray level histogram is a key requirement in the Otsu's thresholding [15]. The advantage is the simplicity in calculating the threshold since the calculation involves 1D intensity data and this helps to reduce the computational processing time in real life application [3]. Due to these advantages, a large number of methods have been proposed to improve the original Otsu's method.

The paper [1], proposed iterative method starts with Otsu's threshold and computes the mean values of the two classes as separated by the threshold. The method separates the image into three classes instead of two as the standard Otsu's method does. The first two classes are determined as the foreground and background and will not be processed further. The third class is denoted as a to-be-determined region that is processed at next iteration and combined with foreground and background to create the final segmentation result. Tests on images show that the new iterative method can achieve better performance than the standard Otsu's method.

The paper [17], proposed the equivalences of Otsu's binarization method, to the search for an optimal threshold that provides the largest absolute t-statistic, and of Otsu's multi-level thresholding method to the search for optimal thresholds that provide the largest F-statistic from one-way analysis of variance, have been stressed. However, these improved Otsu methods are usually unable to produce satisfying segmented results for noised images.

The paper [4], proposed Otsu method with two different approaches which are iteration approach and custom approach. The implementation shows that both of these approaches have given almost the same threshold value for segmenting image. The maximum result for between class variance of gray levels is defined as custom approach while the minimum result within class variance is defined as iterative approach [13].

Chapter 3
Methodology

3.1 Methodology

The methodology includes implementing the Otsu thresholding and K-means clustering algorithms and testing the algorithms with quality parameter for image segmentation. Various types of quality parameters have been used for the sole purpose of evaluating performance analysis of different type of method used for segmenting an image. The details of data collection are discussed in section 4.3.

3.2 Algorithms

3.2.1 Image segmentation using Otsu thresholding

In image segmentation using the Otsu thresholding the gray scale image is given as an input and if the image is color, color to gray-scale conversion is done. There are different methods for the color to gray scale conversion of an image. Here luminosity method is used for the purpose of color to gray-scale conversion. The luminosity method forms a weighted average to account for human perception. Human perceive green more strongly than red, and red more strongly than blue. A commonly used formula for the conversion is:

Gray = 0.21 R + 0.72 G + 0.07 B.

The Otsu method is based on the assumption that the image has a bimodal histogram and there are two distinguishable peaks one for the foreground object and one for the background object. The mean and variance of the background and foreground is then computed. The between class variance of the image is computed for threshold value presents in the image histogram. The optimum threshold value is the value for which the value of between class variance is maximum. The computational formula for the mean, variance and within class variance and between class variance is described in section 1.1.2. All the pixels value greater or equal to the threshold will be measured as 1 and other values as 0. So, the thresholded binarized image is segmented image. The algorithm for the Otsu thresholding is summarized below:

i. Figure out histogram and probabilities of each intensity level

ii Compute initial class's probability $\omega_i(t)$ and initial class class's mean $\mu_i(t)$

iii. Rank through entire possible thresholds ranging from t = 1 to maximum intensity

i. Update ω_i and μ_i

ii. Compute $\sigma_b^2(t)$

iv. Desired threshold corresponds to the maximum $\sigma_b^2(t)$

Process Flowchart

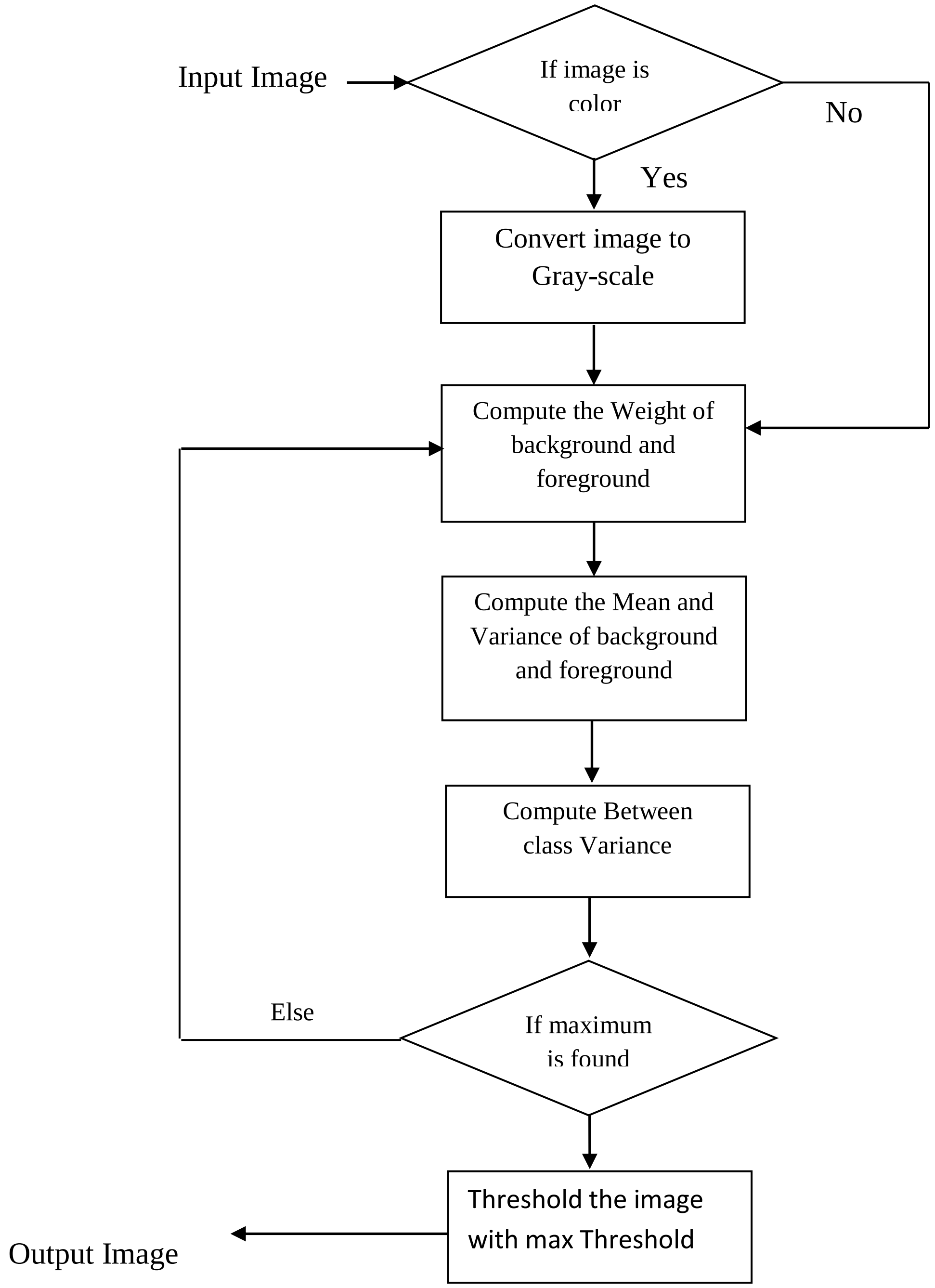


Figure 1: Image Segmentation Using Otsu Thresholding

3.2.2 Image segmentation using K-means algorithm

In image segmentation using the K-means an image is given as input. The number of cluster and centroid is initialized. For every pixel in an image the distance from each centroid is computed. The pixel is assigned to the cluster for which the distance from centroid is minimized. Once the clustering recalculates the centroid value and again measure the distance of every pixel from new centroid, based on the minimum distance from the centroid assign the pixel to the cluster. This iteration will be continued until the centroid does not changes or the pixel values remains in the same cluster in the next iteration. At the end the newly formed cluster pixels are reshaped into the image. Here, two cluster are used for the experiment (k=2) and the images become more clear as the number of the cluster increases.

If an image (x, y) which is to be used for clustering as to form k number of clusters and p(x, y) be a pixel which is to be clustered and C_k be the centroid. The algorithm for K-means clustering is as follows:

i. Initialize the number of clusters k and centroid C_k.
ii. For every pixel, calculate the distance between each pixel and centroid.
iii. On the basis of distance d, allocate the all pixels to nearest centroid.
iv. After assigning all the pixels, recalculate the centroid C_k.
v. Repeat the step i to iv until the pixel values does not change in the next iteration.
vi. Reshape the cluster pixels into image.

Process Flowchart

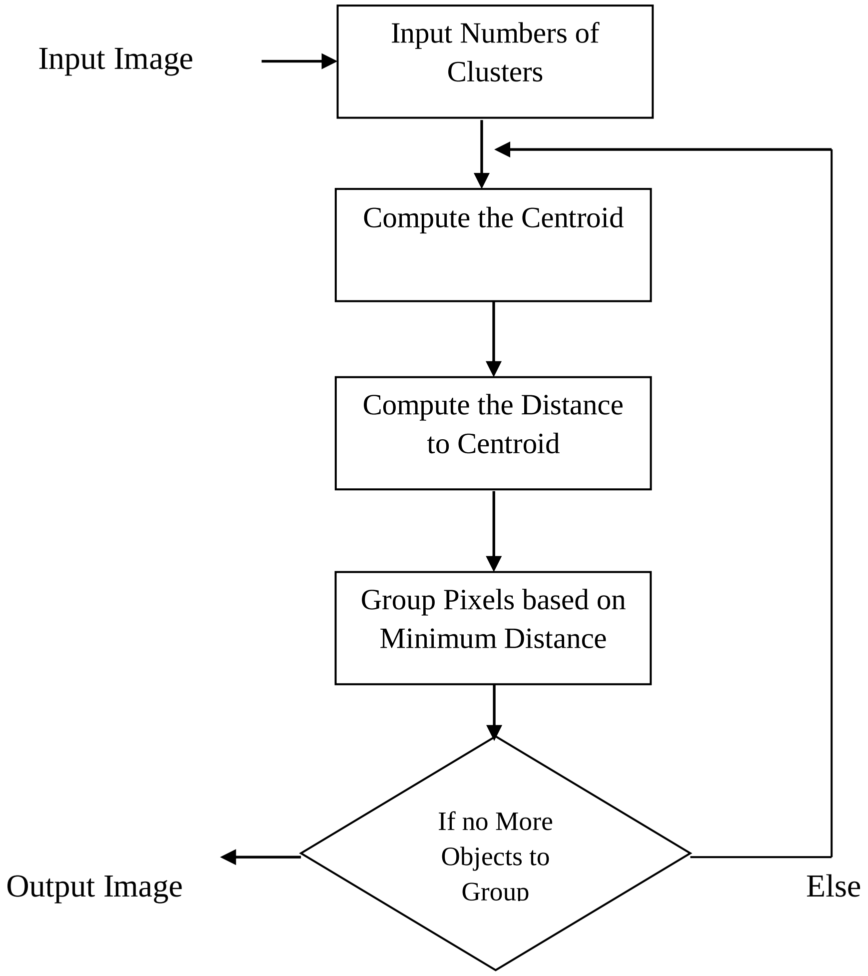


Figure 2: Image Segmentation Using K-Means Clustering

Chapter 4

Implementation and Result Analysis

4.1 Testing Environment

The algorithm has been implemented using following software and hardware configuration.

Hardware Specification:

- Device : Laptop
- System : Intel(R) Core(TM) i5-3337U CPU @1.80GHz
- Hard Disk : 500 GB
- RAM : 4GB

Software Specification:

- Operating System : Windows 10 Pro
- Language : Java
- Tools : Eclipse Java 2019-06 and Image J

4.2 Implementation Tools

All the implementation is done in Java programming language using Eclipse IDE 2019-06

4.2.1 Java Programming Language

Java is a programming language that is class-based, object-oriented, and designed to have as few implementation dependencies as possible. Java was originally developed by James Gosling at Sun Microsystems and has since been acquired by Oracle and released in 1995 as a core component of Sun Microsystems' Java platform. Java applications are typically compiled to byte-code that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture.

Java is one of the most popular and widely used programming languages in use particularly for client-server web applications. It is also a platform that helps to develop run programs written in any programming language

4.2.2 Eclipse IDE 2019-06 and Image J

Eclipse is an integrated development environment (IDE) used in computer programming originally created by IBM in November 2001. In 2004, the Eclipse Foundation was founded to lead and develop the Eclipse community. Eclipse is written mostly in Java and C. Its primary use is for developing Java applications. The Eclipse platform can be used to develop rich client applications, integrated development environments, websites and web services. It includes code editor and has integrated debugger.

Eclipse provide built-in languages such as Ada, ABAP, C, C++, C#, JavaScript, Perl, PHP, Prolog, Python, R, Ruby, Rust, XML, HTML, CSS etc. Users can extend its abilities by installing plug-in written for the Eclipse Platform, such as development toolkits for other programming languages, and can write and contribute their own plug-in modules.

ImageJ is a free open source application to process images. It automates tasks and creates custom tools using macros. ImageJ is used as an image processing toolkit to develop applets, servlets, or applications. It uses 8-bit gray-scale or indexed color, 16-bit unsigned integer, 32-bit floating-point, and RGB color data types. It also creates rectangular, elliptical, or irregular area selections. Supports smoothing, sharpening, edge detection, median filtering, and thresholding on both 8-bit gray-scale and RGB color images. Measure the area, mean, standard deviation, min, and max of entire image. It can also split a 32-bit color image into RGB or HSV components and merge 8-bit components into a color image.

4.3 Data Collection

The input data for the experiment are the different standard image like Lena, Cameraman, etc. of different dimension as 256*256 and 512*512 collected from Berkeley segmentation dataset[1], Kaggle image dataset[2], and Sipi images dataset[3] in JPG format. Any image format like jpg, tif, png etc. can be used as input images.

4.4 Performance Evaluation Parameters

Various types of quality parameters are used for the purpose of evaluating performance analysis of mentioned algorithm used for segmentation with different images. The performance of image segmentation algorithms is evaluated in terms of Peak Signal to Noise Ratio (PSNR), Mean Square Error (MSE), Structural Similarity Index (SSIM), and Structural

[1] https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/
[2] https://www.kaggle.com/ruizgara/socofing/
[3] https://sipi.usc.edu/database/

Content (SC) [7]. Performance analysis helps to seek out the most efficient technique/method used for segmenting an image by thoroughly analyzing the used parameters values. A parameter values can be either high or low in accordance to the type of parameter used.

4.4.1 Peak Signal to Noise Ratio (PSNR)

PSNR is usually measured as decibel scale. In general, the PSNR is used to determine the quality of rebuilding of the image. It measures the quality of image between original image and segmented image. The PSNR must have a higher value to have a higher quality segmented image. It is defined as:

$$\text{PSNR} = 10 \log_{10} \frac{MAX^2}{MSE}$$

Where, MSE is the mean square error, MAX is the maximum possible pixel value of the image. The PSNR must have a higher value to have a higher quality segmented image.

4.4.2 Mean Square Error (MSE)

MSE measures the average of the squares of errors, that is, the difference between the estimator and what is estimated. Mean Square Error (MSE) is calculated pixel-by-pixel by adding up the squared difference of all the pixels and dividing by the total pixel count. MSE of the segmented image can be calculated by using the equation given below:

$$\text{MSE (OI, SI)} = \frac{\left(\sum_{i=0} \sum_{j=0} [\text{OI}(i,j) - \text{SI}(i,j)]^2\right)}{\text{MN}}$$

Where, M and N are the number of rows and columns in the input image respectively, while OI and SI is the original and segmented image. MSE must have a lower value to have a higher quality segmented image.

4.4.3 Structural Similarity Index (SSIM)

SSIM measures the similarity between two images. The SSIM index is a full reference metric; in other words, the measuring of image quality based on an initial uncompressed or distortion-free image as reference. SSIM is calculated using the following equation:

$$\text{SSIM (x,y)} = \frac{(2\mu_x\mu_y + c1)(2\sigma_{xy} + c2)}{(\mu_x^2 + \mu_y^2 + c1)(\sigma_x^2 - \sigma_y^2 - c2)}$$

Where μ, σ, σ_{xy} mean, variance and covariance of image and c1 and c2 are stabilizing constants: c1 $=(K_1 L)^2$, c2 = $(K_2 L)^2$ L= dynamic range of the pixel-values; k_1=0.01 and k_2 = 0.03 by default and higher the value of SSIM, higher the similarity between the two image.

4.4.4 Structural Content (SC)

The value of SC to a great extent influences the quality of segmented image. SC measures is given by

$$SC = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} in(i,j)^2}{\sum_{i=1}^{m}\sum_{j=1}^{n} seg(i,j)^2}$$

Where in(i, j)^2 is a the input image and seg (i ,j)^2 is the segmented image and m and n are matrix row and columns respectively. A smaller value of structural content (SC) means that the image is of better quality.

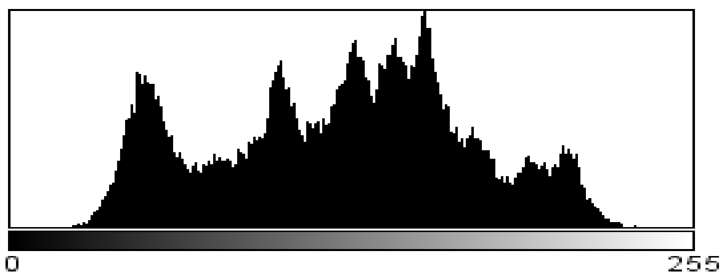
Table 1: Threshold Value Measure

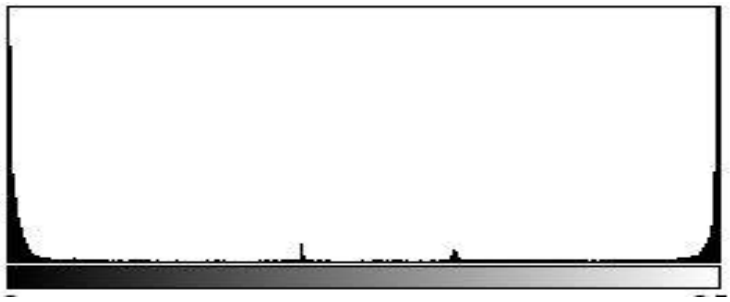
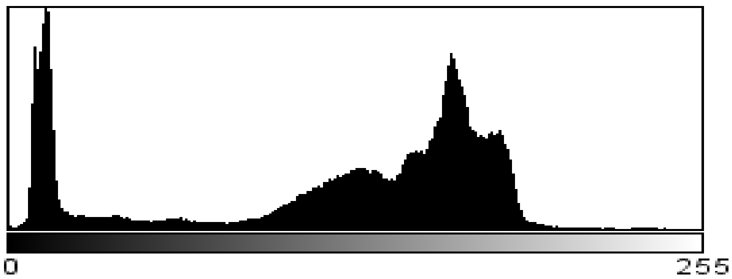
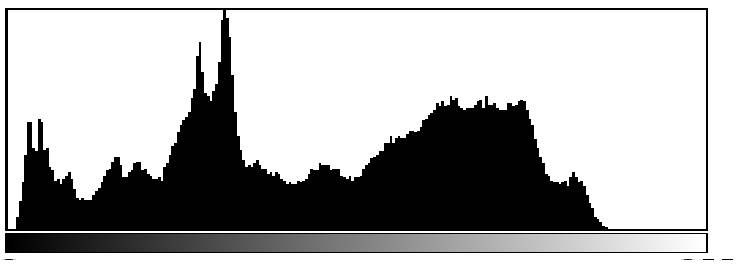
S.N	Image	Dimension	Type	Method	Threshold value Obtained
1.	Lena.jpg	256*256	Gray	Otsu	169
2.	Fingerprint.jpg	256*256	Gray	Otsu	123
3	Cameraman.jpg	512*512	Gray	Otsu	134
4.	Peppers.jpg	512*512	Color	Otsu	115

Table 2: MSE, PSNR, SSIM, SC Measures (No. of Clusters =2)

S.N	Parameter	Method	**Images** (in jpg)				Average
			Lena 256*256 Gray	Fingerprint 256*256 Gray	Peppers 512*512 Color	Cameraman 512*512 Gray	
1	**MSE(dB)**	**Otsu**	7795.24814	2003.3274	6460.9315	3662.9637	4980.618
		K-means	345.5975	1434.887	1232.681	98.05963	777.8064
2	**PSNR**	**Otsu**	9.21251	15.1133	10.0279	12.4975	9.205826
		K-means	22.7451	16.5627	17.2223	28.2160	21.186525
3	**SSIM**	**Otsu**	0.5466	0.9273	0.6744	0.6989	0.7118015
		K-means	0.6843	0.9353	0.7687	0.8960	0.8211
4	**SC**	**Otsu**	0.7978	0.4599	0.6774	0.6528	0.647
		K-means	1.0110	1.0582	1.0896	1.0075	1.0416

Table 3: Image Histogram

S.N	Image	Original Image histogram
1.	Lena.jpg	

2.	Fingerprint.jpg	
3.	Cameraman.jpg	
4.	Peppers.jpg	

4.5 Result Analysis

The results of Otsu and K-means segmentation algorithms has been analyzed by MSE, PSNR, SC, and SSIM parameters value with input image. The histogram of an image is analyzed to evaluate whether the image has the bimodal histogram or not.

The analysis has been done to find the efficient method for the image segmentation with both approaches. The analysis result shows that the average value of MSE for Otsu method is 4980.618 K-means method is 777.8064 respectively. Lower the MSE higher the segmented image quality. So, K-means has the better results.

Also the analysis result shows that the K-means have generated better results with parameter PSNR and SSIM. The average value of PSNR for K-means is 21.186525 and the average value of the PSNR for the Otsu is 9.205826. The PSNR value of the K-means method is higher which shows that the original and segmented image do not loss much of the property.

The SSIM values for Otsu and K-means method are calculated and obtained 0.7118015 and 0.8211 respectively. The SSIM value of the K-means method has higher value than the Otsu method, which shows that the original and the output images have the higher similarities. The SC values for Otsu and K-means method are calculated and obtained 0.647 and 1.0416 respectively. The SC value of the Otsu method has a lower value than the K-means method. The lower SC value of the input and segmented images specifies the better quality of the segmented image.

The binarization of the input image is totally based on the threshold value of image being used. The results also show that input images like Cameraman, Fingerprint have the bimodal histogram and Otsu method can distinguish the foreground and background of the mentioned images rather than rest of images being used. i.e. there are two clear distinguishable peaks for the background and foreground object. Hence, Otsu method does not perform well when there are multiple objects in the image.

Chapter 5
Conclusion and Future Recommendation

5.1 Conclusion

Image segmentation is an important step for image processing and it is used everywhere when the internal part of the image is to be analyzed and is used for dividing the image into different segments and discrete regions. The outcome of image segmentation is a group of segments that jointly encloses the whole image. In this study, Otsu thresholding and K-means has been implemented.

The Otsu's method attempts to find the best threshold value by maximizing between class variance. The K-means clustering algorithm requires that the number of clusters to be predefined and the result varies as the number of cluster changes. From the results obtained it can be concluded that K-means perform better for some parameter than the Otsu method. The higher value of PSNR and SSIM shows that for K-means original and segmented images do not loss the much of the property and have the higher similarities in input and segmented image. The lower SC value of an input and segmented image indicate the better quality of the image is generated by Otsu thresholding.

Also the histogram analysis shows that, for the image with multiple objects and without having bimodal histogram, the Otsu can determine the threshold but cannot detect and separate background and foreground of the image clearly.

5.2 Future Recommendation

The basic Otsu technique is not worthy for noisy images. To overcome this drawback an enhanced form of Otsu can be implemented in future. The detailed study of algorithms to predict the number of clusters in k-means can be performed. Same analysis can be performed by using the distance measure techniques like city block, Manhattan and chessboard distance instead of using and Euclidean distance.

References

[1]. Cai, H., Yang, Z., Cao, X., Xia, W. and Xu, X., A New Iterative Triclass Thresholding Technique in Image Segmentation, *IEEE T*ransactions *on Image Processing,* vol. 23, no. 3, pp.1038-1046, 2014.

[2]. Dhanachandra, N., Manglem, K, and Chanu, Y., Image Segmentation using *K*- means Clustering Algorithm and Subtractive Clustering Algorithm, In Proc. Eleventh International Multi-Conference on Information Processing 2015, pp. 664-671.

[3]. Goh, T., Basah, S., Yazid, H. and Safar, M., Performance analysis of image thresholding: Otsu technique, *Elsevier Ltd.*vol. 114, pp. 298-307, 28 September 2017.

[4]. Kalathiya, S. and Patel, P., Implementation of Otsu Methods with Two Different Approaches, *International Journal of Software Hardware Research Engineering,* vol.2, pp. 2347-4890, 2014.

[5]. Kaur, D. and Kaur, Y., Various Image Segmentation Techniques: A Review, *International Journal of Computer Science and Mobile Computing*, vol. 3, no. 5, pp. 809-814, 2014.

[6]. Kaur, N. and Kaur, R., A review on various methods of image thresholding, *International Journal on Computer Science and Engineering*, vol. 3, no. 10, pp. 3441-3443, 2011.

[7]. Kongkham, G. and Kumar, S., Analysis of Image Segmentation Techniques, *International Research Journal of Computer Science,* vol. 2, no. 6, pp.45-53, 2015.

[8]. Lin, WT., Lin CH., Wu, TH. and Chan, YK., Image Segmentation Using the K-means Algorithm for Texture Features, *International Journal of Computer, Electrical, Automation, Control and Information Engineering,* vol. 4, no. 5, pp. 945-948, 2010.

[9]. Patil, A. and Shaikh, J., Otsu Thresholding Method for Flower Image Segmentation, *International Journal of Computational Engineering Research*, vol. 6, no. 5, pp. 1-6, 2016.

[10]. Purohit, P. and Joshi, R., A New Efficient Approach towards k-means Clustering Algorithm, *International Journal of Computer Applications,* vol. 65, no. 11, 2013.

[11]. Qu, Z. and Zhang, L., Research on Image segmentation Based on the Improved Otsu Algorithm, presented at Second International Conference on Intelligent Human-Machine Systems and Cybernetics, China, 2010.

[12]. Saini, S. and Arora, K., A Study Analysis on the Different Image Segmentation Techniques, *International Journal of Information and Computation Technology,* vol. 4, no. 4, pp. 1445-1452, 2014.

[13]. Sezgin, M. and Sankur, B., Survey Over Image Thresholding Techniques and Quantitative performance evaluation, *Journal of Electronic Imaging*, vol. 13, pp. 146-165, 2004.

[14]. Tokas, N., Karkra, S. and Pandey, M., Comparison of Digital Image Segmentation Techniques- A Research Review, *International Journal of Computer Science and Mobile Computing,* vol. 5, no. 5, pp. 215-220, 2016.

[15]. Vala, H. and Baxi, A., A Review on Otsu Image Segmentation Algorithm, *International Journal of Advanced Research in Computer Engineering & Technology,* vol. 2, no. 2, pp. 387-389, 2013.

[16]. Vats, I. and Singh, K., Otsu Image Segmentation Algorithm: A Review, *International Journal of Innovative Research in Computer and Communication Engineering,* vol. 5, no. 6, pp. 11945-11948, 2017.

[17]. Xue, H. and Titterington, M., t- Tests, F-Tests and Otsu's Methods for Image Thresholding, *IEEE T*ransactions *on Image Processing,* vol.20, no.8, pp.2392-2396, 2011.

[18]. Yedla, M., Rao, S. and Srinivasa, T., Enhaning K-means Clustering Algorithm with Improved Initial Center, *International Journal of Science and Information Technologies*, vol. 1, no. 2, pp. 121–125, 2010.

Appendix A

Lena Image

Lena Image Segmented using Otsu

Lena Image Segmented using K-means (k=2)

Lena Image Segmented using K-means (k=3)

Fingerprint Image

Fingerprint Image Segmented using Otsu

Fingerprint Image Segmented using K-means (k=2)

Fingerprint Image Segmented using K-means (k=3)

Cameraman Image

Cameraman Image Segmented using Otsu

Cameraman Image Segmented using K-means (k=2)

Cameraman Image Segmented using K-means (k=3)

Peppers Image

Peppers Image Segmented using Otsu

Peppers Image

Peppers Image Segmented using K-means (k=2)

Peppers Image Segmented using K-means (k=3)

Appendix B

Source Code

OtsuBinarize.java

```
publicclass OtsuBinarize {

    privatestatic BufferedImage orgimage, grayimage, binimage;

    publicstaticvoid main(String[] args) throws IOException {

        orgimage = ImageIO.read(orgimage_f);
        grayimage = toGray(orgimage);
        binimage = binarize(grayimage);
        writeImage(output_f);

    }
    publicstaticint[] imageHistogram(BufferedImage input) {

        int[] histogram = newint[256];

        for (int i = 0; i < histogram.length; i++)
            histogram[i] = 0;

        for (int i = 0; i < input.getWidth(); i++) {
            for (int j = 0; j < input.getHeight(); j++) {
                int red = new Color(input.getRGB(i, j)).getRed();
                histogram[red]++;
            }
        }

        return histogram;

    }
BufferedImage lumino = new BufferedImage(orgimage.getWidth(), orgimage.getHeight(),
orgimage.getType());

        for (int i = 0; i < orgimage.getWidth(); i++) {
            for (int j = 0; j < orgimage.getHeight(); j++) {

                red = new Color(orgimage.getRGB(i, j)).getRed();
                green = new Color(orgimage.getRGB(i, j)).getGreen();
                blue = new Color(orgimage.getRGB(i, j)).getBlue();

                red = (int) (0.21 * red + 0.72 * green + 0.07 * blue);
                newPixel = colorToRGB(red, green, blue);
                lumino.setRGB(i, j, newPixel);

            }
```

```
            }

            return lumino;
    }

BufferedImage binimage = new BufferedImage(orgimage.getWidth(),
orgimage.getHeight(), orgimage.getType());

            for (int i = 0; i < orgimage.getWidth(); i++) {

                for (int j = 0; j < orgimage.getHeight(); j++) {


                    red = new Color(orgimage.getRGB(i, j)).getRed();

                    if (red > threshold) {

                        newPixel = 255;

                    } else {

                        newPixel = 0;

                    }
                    newPixel = colorToRGB(newPixel, newPixel, newPixel);
                    binimage.setRGB(i, j, newPixel);

                }
            }

            return binimage;

    }       privatestaticint otsuTreshold(BufferedImage orgimage) {

for (int t = 0; t < 256; t++) {
                    wB += histogram[t];
                    if (wB == 0)
                        continue;
                    wF = total - wB;

                    if (wF == 0)
                        break;

sumB += (float) (t * histogram[t]);
float mB = sumB / wB;
float mF = (sum - sumB) / wF;
float varBetween = (float)wB * (float)wF * (float)(mB - mF) *(float (mB - mF);
        if (varBetween > varMax) {
            threshold = t;
            varMax = varBetween;
```

```
                return threshold;
}
```

Kmeans.java

```
public class KMeans{
public static void main(String[] args) {
public Cluster findMinimalCluster(int rgb) {
    Cluster cluster = null;
int min = Integer.MAX_VALUE;
for (int i=0;i<clusters.length;i++) {
int distance = clusters[i].distance(rgb);
if (distance<min) {
        min = distance;
        cluster = clusters[i];
      }

Void removePixel(int color) {
int r = color>>16&0x000000FF;
int g = color>> 8&0x000000FF;
int b = color>> 0&0x000000FF;
      reds- =r;
      greens- =g;
      blues- =b;
      pixelCount--;
      red   = reds/pixelCount;
      green = greens/pixelCount;
      blue  = blues/pixelCount;


    }
public Cluster(int id, int rgb) {

int r = rgb >>16&0x000000FF;
int g = rgb >> 8&0x000000FF;
int b = rgb >> 0&0x000000FF;
      red = r;
      green = g;
      blue = b;
this.id = id;
      addPixel(rgb);
      return cluster;

}

void addPixel(int color)
            {
int r = color>>16&0x000000FF;
int g = color>> 8&0x000000FF;
int b = color>> 0&0x000000FF;
```

```
            reds+=r;
            greens+=g;
            blues+=b;
            pixelCount++;
            red   = reds/pixelCount;
            green = greens/pixelCount;
            blue  = blues/pixelCount;
        }

int istance(int color) {
int r = color>>16&0x000000FF;
int g = color>> 8&0x000000FF;
int b = color>> 0&0x000000FF;
int rx = Math.abs(red-r);
int gx = Math.abs(green-g);
int bx = Math.abs(blue-b);
int d = (rx+gx+bx) / 3;
return d;

        }
}
```