ISSN: 0976-9102 (ONLINE)
DOI: 10.21917/ijivp.2022.0392

IMAGE SEGMENTATION USING MULTI-THRESHOLD TECHNIQUE BY HISTOGRAM SAMPLING

Sangyal Lama Tamang and Amit Gurung
Department of Information Technology, Martin Luther Christian University, India

Abstract

The segmentation of digital images is one of the essential steps in image processing or a computer vision system. It helps in separating the pixels into different regions according to their intensity level. A large number of segmentation techniques have been proposed, and a few of them use complex computational operations. Among all, the most straightforward procedure that can be easily implemented is thresholding. In this paper, we present a unique heuristic approach for image segmentation that automatically determines multilevel thresholds by sampling the histogram of a digital image. Our approach emphasis on selecting a valley as optimal threshold values. We demonstrated that our approach outperforms the popular Otsu's method in terms of CPU computational time. We demonstrated that our approach outperforms the popular Otsu's method in terms of CPU computational time. We observed a maximum speed-up of 33.63× and a minimum speed-up of 10.21× on popular image processing benchmarks. To demonstrate our approach's correctness in determining threshold values, we compute PSNR, SSIM, and FSIM values to compare with the values obtained by Otsu's method. This valuation shows that our approach is comparable and better in many cases than well-known Otsu's method.

Keywords:

Digital Image Processing, Image Segmentation, Multilevel Thresholding, Histogram, Histogram Valley

1. INTRODUCTION

In most computer vision systems, one of the essential preprocessing tasks is the image segmentation. The reliability of the outputs depends on the quality of the input image provided by the image preprocessing. Thus, research is progressing in the direction of enhancing the quality of input images to eliminate noise, visual artifacts, and redundancy of information. One of the most used techniques to handle these issues is Image Segmentation. It is the process of grouping pixels into different groups or segments in an image. Each such group represents an object in an image providing a better understanding of the objects in the given image. Recently, image segmentation has been applied in a number of areas such as Medical Imaging for the detection of brain tumor or the study of brain development of neonatal brain from Magnetic Resonance Imaging (MRI) scanning [1]-[4], improvement of irregularity detection in biometric fingerprint [5], landscape analysis of remotely sensed satellite images [6], also for object detection in still and moving images [7].

The approach of image segmentation can be broadly categorized into discontinuity-detection and similarity-detection based methods [8]. The former is an approach of segmenting an image into regions based on discontinuity, whereas the later segments image into regions based on the similarity of pixels. Image segmentation can be achieved by a number of varying techniques, some of these are 1) thresholding, 2) clustering-based, 3) edge-based, 4) region-based, 5) watershed-based methods, 6) partial differential equation-based and 7) Artificial Neural Network (ANN)-based segmentation methods.

One of the simplest image segmentation technique is thresholding. In this method, a threshold value is chosen to segment an image. All pixel values above or below the threshold value are classified as object or as a background. When only a single threshold value is used to segment image, it is known as global thresholding, and when multiple threshold values are used to segment one or more objects, it is referred to as local thresholding techniques. Clustering in image segmentation is a technique of thresholding, in which an image is partitioned into K-clusters. For each K-clusters, a cluster center is chosen randomly (or using a heuristic method). A pixel is assigned to a particular cluster based on the minimum distance between the pixel and the cluster center. The distance metric is usually based on features such as pixel color, intensity, texture, etc. These processes are iterated to compute appropriate cluster centers until convergence is achieved. Segmentation of image is achieved by mapping these clusters back to the original spatial domain [9]. However, edge-based image segmentation is based on the theory that segmentation can be achieved by detecting discontinuity of pixels lying on the boundary between different regions. Gray histogram and gradient based method are two main edge-based segmentation methods [10]. The region-based segmentation approach is based on the partitioning of the image into different regions according to a set of predefined criteria [11]. Another segmentation method, watershed-based image segmentation replicates the process of rainfall in a real landscape. In a gray-scale landscape, light and dark intensity pixel are considered as hills and hollows of a gray-scale image. When an imaginary rainfall occurs in a gray-scale landscape, the rain flows from high altitude (gray level area) to some low lying (gray level) region. This flow creates watersheds or catchment basins. A gray-scale landscape is then segmented or partitioned into regions according to watersheds [12]. When looked into a supervised segmentation along with a training data set a little or incomplete knowledge of the problem is required. Artificial Neural Networks (ANN) is a technique of supervised image segmentation. ANN are networks of interconnected parallel processing units. ANN partitions the image into multiple segments where all pixels in a partition holds some similar characteristics. As any other supervised learning model, ANN can learn by examples [13]. Extracting the desired object of interest from an image has always been the fundamental and most important task in image segmentation. When considering partial differential equations (PDE) for image segmentation, PDE always considers images as continuous objects. Due to the flexible structure, PDE converts images into initial and boundary conditions and later obtains the segmentation result as the solution of the equation [14].

A thresholding-based method is considered to be the simplest among all the known techniques. The problem is to determine a

threshold value (for global thresholding) that divides the pixels into different classes. Two famous classic works are attributed to Otsu [15] and Kapur et al. [16]. The core idea behind Otsu's method is to maximize the between-class variance of gray levels. Kapur et al. [16] propose the maximization of histogram entropy of segmented classes to select the optimal threshold value. Both these methods (Otsu's and Kapur's) can be easily extended for multilevel thresholding. However, they are inefficient in determining optimal thresholds due to the exponential growth in the computational complexity of the algorithm. The precision of the algorithm also decreases as the number of thresholds increases [17].

An approach that is in line with Kapur's work is minimization of cross entropy commonly referred to as MCET. The work was initially presented by Solomon Kullback in [18]. MCET was considered as an extension to Kapur's work. However, due to the computational complexity of determining the optimal threshold, the problem remains. To address this problem, researchers propose a large number of meta-heuristic optimization algorithms. Some of these optimization algorithms minimizes the cross-entropy [19]-[22], while others maximize the Kapur's entropy [17], [23], [24], [25] (or maximize Otsu's between class variance [17], [24], [25]) to determine optimal threshold values. Segmentation of image is one of the most essential and preliminary steps in many applications related to computer vision and image processing. These meta-heuristic optimization algorithms converge to the optimal solution faster than the exhaustive search. However, when the dimension of the problem (i.e., the number of thresholds to be found out) increases, there is a proportional increase in the search time. The problem becomes worst for an image having higher dimensions (image size).

Image segmentation based on the histogram of an image is a popular thresholding technique. A histogram of an image consists of a number of peaks and valleys, and each valley separates a region or an object from its background. When there are only two distinct peaks in a histogram, it forms a bi-modal histogram. A valley between the two peaks forms an optimal global threshold value. However, when more than two peaks exist, global thresholding may not serve well. It requires more than one threshold value or a multilevel thresholding technique is a need. In this paper, we propose a heuristic method of image segmentation using the multi-threshold technique by sampling the histogram of a digital image. Our algorithm is designed for a gray-scale image of n-levels. The algorithm consists of three main steps. First, it iterates over the n-levels and determines all valleys from the histogram so as to emphasis the resultant threshold as valley [26]. Second, the histogram is equally partitioned into r-regions and determine points having minimum value (Frequency in a histogram see Fig.1) within each region. The goal of this step is to select the minimum point within a region. The only two possibilities for this point is the lowest valley or a descending slope in the region. The advantage of this step is two-fold 1) it helps to eliminate a local minima problem within a region, which is a serious issue in most optimization algorithm and 2) it helps to select threshold point in a uniformly distributed fashion. Finally, the third step is to choose these optimal threshold values obtained in the previous two steps. Candidate points are formed by choosing common points in the two prior steps (valley points from the first step and minimum points from the second step). We adopt an ad-hoc approach of clustering the candidate points and select a mean of the cluster as an optimal value. The number of clusters is the number of threshold values to be determined. To emphasis valley as the threshold, we select the immediate next candidate point to the mean of the cluster. The implementation of this approach and benchmarks reported in the paper can be downloaded from https://sites.google.com/view/imagesegmentation/downloads.

The rest of the paper is organized as follows. Section 2 provides the necessary basic concepts in image processing. In Section 3, we present our algorithm of multilevel thresholding. In Section 4, we provide the experimental results to illustrate the performance of our approach compared to the most popular multilevel thresholding method. We conclude in Section 5.

2. PRELIMINARIES

We propose an algorithm for image segmentation using the multi-thresholding technique for digital images. Our approach is mainly based on the histogram generated from the gray-scale image of the given image.

2.1 DIGITAL IMAGE

Digital images are two-dimensional (2D) images defined as some function $f(x,y)$, where x and y are known as spatial or plane coordinates. Digital images are transformed images from analog media to electronic data which can be saved, organized, retrieved and restored through electronic devices [27]. They can be broadly classified into three different types on the basis of their size and range of pixel values as:

- **Binary Image**: images with only two possible values for every pixel are binary images. Each pixel will be stored as a single bit, i.e., 0 or 1.
- **Gray-scale Image**: Gray-scale images are 8-bit images giving a possible range of pixel values from $L \in [0, 255]$. The pixel values in a gray-scale image represent the brightness of the pixel. Typically, zero is taken to be black, and 255 is taken to be white.
- **Color Image**: It is also known as an RGB image, where R, G, and B stands for the primary color red, green, and blue, respectively. The RGB image is a system for representing the color to be used in a computer display as a two-dimensional array of small integers. Each of these integers represents a pixel value for an image. An RGB image has three-pixel values, one for each of red, green, and blue colors.

Color images are converted to equivalent gray-scale using the standard formula [28]

$$I_{gray}(i,j) = \begin{bmatrix} 0.2989 & 0.5870 & 0.1140 \end{bmatrix} \times \begin{bmatrix} R(i,j) \\ G(i,j) \\ B(i,j) \end{bmatrix} \quad (1)$$

where $R(i,j)$, $G(i,j)$ and $B(i,j)$ are respectively red, green and blue pixel values of the color image. $I_{gray}(i,j)$ is the equivalent gray-scale value computed as a weighted sum of these three components. These gray-scale images, in turn, can be easily converted to a binary image by applying a global thresholding technique.

$$I_{bw}(i,j)=\begin{cases}1 & if\ I_{gray}(i,j)<th\\ 0 & f\ I_{gray}(i,j)\geq th\end{cases} \quad (2)$$

where, *th* is the chosen global threshold value and $I_{bw}(i,j)$ is the corresponding binary value generated for the selected threshold for the image.

2.2 IMAGE HISTOGRAM

For visualization of a target object from the background image, the histogram-based thresholding technique is the most commonly used approach for image segmentation, in digital image processing [29]. A histogram is a graph consisting of x- and y-axis, where the x-axis is the gray level pixel values, and the y-axis gives the number of pixels (or frequency) corresponding to the gray levels. The Fig.1 shows a histogram plot of the gray-scale image of Lena. The histogram plot is a nonlinear curve. In this paper, we call a peak to a point representing the highest frequency in a curve and a valley to the point denoting the least frequency in a curve (Fig.1).

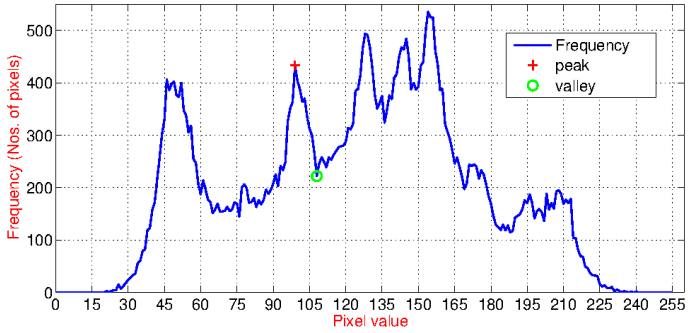

Fig.1. Histogram of a gray-scale image of Lena. The symbol + (colored in red) and o (colored in green) shows a peak and a valley point in a curve.

2.3 MULTILEVEL THRESHOLDING

Determining the best threshold for any digital image is computationally expensive. The three most popular methods found in the literature for global thresholding that is subsequently extended for multilevel thresholding are presented in the following subsections.

2.3.1 Maximizing Variance between Classes:

Otsu proposes to maximize the between class variance in order to determine the best threshold value for image segmentation [15] (object from the background). Let h represent the histogram of the image in gray-scale such that $h(i) = n_i/N$ and $N = n_1+n_2+...+n_L$ where n_i is the number of pixels in level *i*. Then, the probabilities of class occurrence [1, ..., (*th*-1)] and [*th*, ..., *L*] are given by

$$\omega_0(th)=\sum_{i=1}^{th-1}h(i) \text{ and } \omega_1(th)=\sum_{i=th}^{L}h(i) \quad (3)$$

whereas the class mean is represented by

$$\mu_0(th)=\sum_{i=1}^{th-1}\frac{ih(i)}{\omega_0} \text{ and } \mu_1(th)=\sum_{i=th}^{L}\frac{ih(i)}{\omega_1} \quad (4)$$

Therefore, to determine the best threshold value, *th* Otsu proposes to maximize the between-class variance denoted by

$$\sigma_B^2(th)=\omega_0\omega_1(\mu_1-\mu_0)^2 \quad (5)$$

This method can be easily extended to support multilevel thresholding. For instance, when the number of thresholds to be determine is two (th_1,th_2), three classes or probability distributions are formulated as X_1 = [1, ..., (th_1-1)], X_2 = [th_1, ..., (th_2-1)] and X_3 = [th_2,...,*L*]. Accordingly, the optimal thresholds is now a function of two variables th_1, th_2 which is computed as

$$\arg\max\left\{\sigma_B^2(th_1,th_2)\right\} \quad (6)$$

2.3.2 Maximizing Entropy:

Kapur et al. [16] presented an algorithm based on the concept of entropy to segment digital image. Let h(i) bears the same meaning as in Equation 3. To determine two thresholds (say t = [th_1, th_2]), where 1 < th_1 < th_2 < L, the probabilities of class occurrence [1, ..., (th_1-1)], [th_1,...,(th_2-1)] and [th_2, ..., *L*] are given by

$$\omega_0(t)=\sum_{i=1}^{th_1-1}h(i);\ \omega_1(t)=\sum_{i=th_1}^{th_2-1}h(i);\ \omega_2(t)=\sum_{i=th_2}^{L}h(i) \quad (7)$$

Then, the entropies for each of these classes are given by

$$H_0(t)=-\sum_{i=1}^{th-1}\frac{h(i)}{\omega_0}\ln\left(\frac{h(i)}{\omega_0}\right)$$
$$H_1(t)=-\sum_{i=th_1}^{th_2-1}\frac{h(i)}{\omega_1}\ln\left(\frac{h(i)}{\omega_1}\right)$$
$$H_2(t)=-\sum_{i=th_2}^{L}\frac{h(i)}{\omega_2}\ln\left(\frac{h(i)}{\omega_2}\right) \quad (8)$$

The optimal thresholds are computed by maximizing the sum of the entropies. For the global threshold, the optimal threshold is given by $\sigma_W(th_1)=H_0(t)+H_1(t)$ here $th_2=L$. Multilevel thresholding, t = [th_1,th_2] optimal thresholds are obtained by maximizing $\sigma_W(th_1,th_2) = H_0(t) + H_1(t) + H_2(t)$.

2.3.3 Minimizing Cross Entropy:

The cross-entropy between two probabilistic distribution is the measure of the statistical difference in uncertainty in the outcome of the experiment when data is transmitted from one distribution to another. Kullback's cross-entropy is given as [18]:

$$\varphi(X,Y)=\sum_{i=1}^{N}x_i\log\left(\frac{x_i}{y_i}\right) \quad (9)$$

where X = {x_1,x_2,...,x_N} and Y = {y_1,y_2,...,y_N} are the two probability distribution. A high value of φ represents more uncertainty in the distribution process.

In a digital image, to segment an image into an object and a background (i.e., into two partitions), a threshold value *th* is chosen. For efficient image segmentation, an optimal threshold is computed by minimizing the cross-entropy given by [30]:

$$\eta(th)=\sum_{i=1}^{th-1}ih(i)\log\left(\frac{\sum_{i=1}^{th-1}ih(i)}{\sum_{i=1}^{th-1}h(i)}\right)+\sum_{i=th}^{L}ih(i)\log\left(\frac{\sum_{i=th}^{L}ih(i)}{\sum_{i=1}^{L}h(i)}\right) \quad (10)$$

where h is the histogram of the image. The computational complexity for determining a single threshold value is $O(L^2)$. However, this complexity increases to $O(L^{n+1})$ for 'n' threshold values. To compute an optimal threshold using exhaustive search, we compute Eq.(10) for all $th \in [1, L]$, and select the th where $\eta(th)$ is the minimum of all. This is computationally an expensive operation, to reduce this complexity, [31] presented an improvement by introducing recursive programming to support multilevel thresholding. Let $[th_1, th_2, ..., th_n]$ be the set of thresholds to be determined, $th_0 < th_1 < th_2, ..., th_n < th_{n+1}$, where $th_0 = 1$ and $th_{n+1} = L + 1$ are dummy thresholds introduced for convenience. The objective function to be minimize, to obtain an optimal threshold can be represented as:

$$\eta\left(th_1, th_2, ..., th_n\right) = \sum_{i=1}^{n+1} m^1\left(th_{i-1}, th_i\right) \log\left(\frac{m^1\left(th_{i-1}, th_i\right)}{m^0\left(th_{i-1}, th_i\right)}\right) \quad (11)$$

where m_0 and m_1 are the values of zero-moment and first-moment points computed as

$$m^0\left(a,b\right) = \sum_{i=a}^{b-1} h\left(i\right)$$

$$m^1\left(a,b\right) = \sum_{i=a}^{b-1} ih\left(i\right) \quad (12)$$

The Eq.(11) reduces the computational complexity from $O(L^{n+1})$ to $O(L^n)$, 'n' being the number of thresholds to be determined. However, this reduction did not do any better when 'n' is significant. The literature presents a large number of meta-heuristic optimization algorithms (mentioned earlier in Section 1). This algorithm applies heuristic techniques on these three popular thresholding methods to converge to the optimal solution in fewer iterations. However, there is no clear winner in this race. Segmentation is an essential step in all image processing and computer vision. Therefore, determining an efficient segmentation technique is still a recent research area in digital image processing.

3. MULTILEVEL THRESHOLDING USING HISTOGRAM SAMPLING

We propose an approach of determining multiple threshold values from a given image represented as a gray-scale image. When the input image is a color image, it is converted into gray-scale using Eq.(1).

3.1 PROPOSED ALGORITHM

We present below our proposed algorithm as five major steps:

Step 1: We obtain the normalized histogram h of the input image represented as $h(i) = n_i/N$ for $N = n_1+n_2+...+n_L$, where n_i is the number of pixels in level i. L is the pixel of an image representing the highest gray level intensity.

Step 2: Let X_y be the set of all valley points between $[1, L]$. A valley point is a pair $(i,h(i))$. We scan the histogram of the image and obtain all pixel-level that represents a valley in the histogram and construct **setA** as:

for i: 1 to L

setA = $\{i: h(i) \in X_y\}$

Note that as mentioned in Section 2.2, a valley is a point representing the lowest frequency of a pixel-level in a curve. A simple method to determine all valleys in the histogram is a gradient search method. In this case, gradient descent is used to find all discrete local minima.

Step 3: We sample the histogram into r-regions or partitions of equal size, and determine pixel-level having the lowest frequency in each of these partitions. We decide the number of partitions as

$$r = L/s \text{ for } s \in \{x: (L \,\%\, x) \text{ is zero, and } x > 1\}$$

where % is the modulo division operator. We computed **setB**, to obtain the set of all minimum points in the histogram h, for each r-partitions as follows:

$$\textbf{setB} = \arg_{r_i - 1 \le I < r_i} \min\left\{h\left(i\right)\right\};\ r_i \text{ is the } i^{\text{th}} \text{ partition.}$$

When $L = 256$, the possible values for r are 2/4/8/16 and the partition sizes can be 128/64/32/16. In this paper, we chose 32 equal partitions. However, when the required number of thresholds is 32 or more, 64 or higher partitions size can be selected. The goal of this step is to select the minimum point within a region. Two possibilities for this point are either the lowest valley or the last spot in that region's descending slope. This step helps eliminate a local minima problem within a region. A local minima problem is a severe concern in most optimization algorithms. Secondly, this partitioning of histograms helps to distribute the candidate thresholds in a digital image histogram uniformly.

Step 4: We now create new set **setC** using sets **setA** obtained in Step 2 and **setB** in Step 3. setC contains the elements that are common in both **setA** and **setB**, computed as

$$\textbf{setC} = \textbf{setA} \cap \textbf{setB}$$

Step 5: Thus, **setC** contains the candidate threshold values. Now, based on the number of threshold values to be determined, we can select appropriate thresholds from the candidate set **setC**. Let t be the number of thresholds to be determined. We adopt a very naive approach of grouping the candidate points into t clusters and select the mean of the cluster as optimal value. The number of clusters formed is based on the number of threshold values to be determined. We emphasis thresholds as valley points and select the immediate next candidate point to the mean of the cluster (for the same reason as mentioned earlier).

For computational efficiency, we perform Step 2 and Step 3 under the same scan of the histogram h. Moreover, the histogram of an image is already a probability distribution function (PDF), and our approach do not depend on the normalized histogram. Therefore, we may skip the computation involved in step 1, instead just use the histogram obtained from the input image.

3.2 COMPLEXITY ANALYSIS

The computational time of our algorithm for computing multilevel thresholds is constant. However, for most algorithms, the computational time increases with an increase in the number of thresholds to be determined. Step 1 is the most expensive step in our algorithm, which computes the histogram of an image. In the worst case, the time to compute a histogram is $O(N^2)$, assuming the image's height and width are equal to N. The next two steps are computed in a single for loop of size L, this we do in $O(L)$, where L is the highest intensity level of the pixel. Step 4 compares the elements of sets **setA** and **setB**, this requires

$O(\max(a,b))$, where a and b are the number of elements in the two sets, a, $b < L$. Finally, Step 5 requires at most r iterations which compute the mean of candidate thresholds for each t clusters. This computation requires $O(r)$, where r and t are the number of partitions and thresholds to be determined, respectively. The number of partitions r is usually constant, and $t < r < L$; therefore, an increase in the size of t does not affect the computational time.

3.3 ILLUSTRATION

We illustrate our approach with the help of Fig.3. The algorithm begins by generating a histogram of the image in Step 1. Step 2 generates all the valley points in the histogram and create set Set-A. In the Fig.3, all points represented by the green circle are valley. In Step 3, the algorithm partitions the histogram into equal-partitions, in this example it is divided into 16 partitions. The dotted lines(- -) in magenta colour denotes the partitions. In each of these partitions, the pixel having the least frequency is chosen to form a set setB. These points are marked as cross (x) in red colour. Step 4 determines only those points that are common in both setA and setB and call this set as setC. In the example, we obtain 13 such points when 16 partitions are chosen. Finally, Step 5 returns the required number of threshold values from the set setC based on a very naive clustering approach.

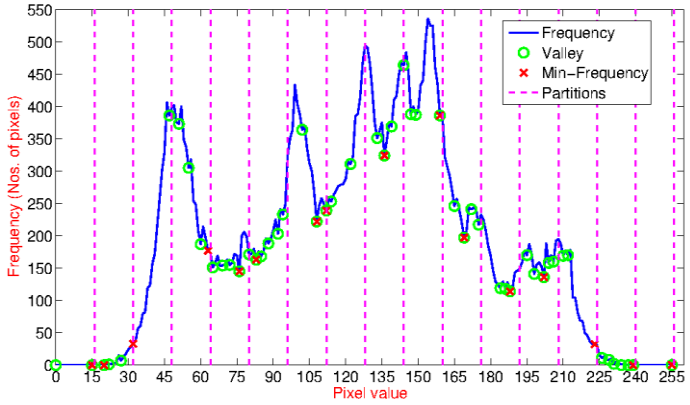

Fig.2. Illustration of our approach on the image of Lena.

3.4 SEGMENTATION USING MULTI-THRESHOLDING

We segment the image into multiple segments by using threshold values obtained from the proposed algorithm. In this work, we use the following approach to segment image into three classes using two thresholds:

$$I_{seg}(i,j)=\begin{cases} I_{gray}(i,j) & if\ I_{gray}(i,j)\le th_1 \\ th_1 & if\ I_{gray}(i,j)\le th_2 \\ I_{gray}(i,j) & if\ I_{gray}(i,j)>th_2 \end{cases} \quad (12)$$

When the number of thresholds is more than two, we use the following approach to generate segmented image:

$$I_{seg}(i,j)=\begin{cases} I_{gray}(i,j) & if\ I_{gray}(i,j)\le th_1 \\ th_1 & if\ th_{i-1}<I_{gray}(i,j)\le th_i \\ I_{gray}(i,j) & if\ I_{gray}(i,j)>th_2 \end{cases} \quad (13)$$

where, i=2,3,…,t-1

4. EXPERIMENTS

We have implemented our proposed algorithm in MATLAB. In the text that follows, we refer to our histogram-based algorithm as AMTIS, abbreviating Automatic Multilevel Thresholding for Image Segmentation. We present an evaluation of the algorithm on various standard benchmarks commonly used in the literature. The performance of our proposed algorithm in comparison to the popular Otsu's method is reported. We use MATLAB's built-in function multithresh, which implements Otsu's method of multilevel thresholding [15]. We use standard image benchmarks that are popular in image processing. Some of these images are obtained from the USC-SIPI image database. The image Frozen Franz Josef is taken from https://earthobservatory.nasa.gov /images/76883/frozen-franz-josef-land. Franz Josef is located 600 miles from the North Pole. Ice covers it throughout, even during the summer. The image is a satellite image made from a combination of visible and near-infrared wavelengths. The fingerprint image is taken from http://bias.csr.unibo.it/fvc2000 /download.asp, using the DB1 B.zip and the benchmark file is 101 1.tif.

4.1 RESULTS

We perform the experiments on AMD FX(TM)-6100 Six-Core Processor, 3.3GHz, with 8 GB RAM. The results are an average of 20 runs. The number of thresholds evaluated is 2, 3, 4, and 5 in line with the results presented in the related literature [32]-[34]. We obtain threshold values using our algorithm AMTIS and generate a segmented image. To verify the quality of the segmented image, we compute the peak-to-signal ratio (PSNR), the structure similarity index (SSIM) and feature similarity index (FSIM).

The PSNR is a measure to determine the quality of the reconstructed image (in this case the segmented image I_{seg}) in comparison to the original image (I_{gray}) using the root mean square error (RMSE) as:

$$PSNR = 20\log_{10}\left(\frac{Max_p}{RMSE}\right),(dB)$$

$$RMSE=\sqrt{\frac{\sum_{i=1}^{m}\sum_{j=1}^{n} I_{gray}(i,j)-I_{seg}(i,j)}{m\times n}} \quad (14)$$

where Max_p is the highest intensity value of a pixel. The unit of measurement is in decibel (dB). When the intensity value is represented using 8-bit, Max_p = 255. A higher PSNR value is desired for better quality [35].

SSIM [36] is used to measure the structural similarity between the original image and the segmented image computed using the Eq.(15). Like PSNR, for a better segmentation quality, a higher value of SSIM is desired.

$$SSIM\left(I_{gray},I_{seg}\right)=\frac{\left(2\mu_{I_{gray}}\mu_{I_{seg}}+C_1\right)\left(2\sigma_{I_{gray}}\mu_{I_{seg}}+C_c\right)}{\left(2\mu^2_{I_{gray}}\mu^2_{I_{seg}}+C_1\right)\left(2\sigma^2_{I_{gray}}\sigma^2_{I_{seg}}+C_2\right)}$$

$$\sigma_{I_{gray}I_{gray}}=\frac{1}{N+1}\sum_{i=1}^{N}\left(I_{gray_i}-\mu_{gray_i}\right)\left(I_{seg_i}-\mu_{I_{seg}}\right) \quad (15)$$

where $\sigma_{I_{gray}I_{gray}}$ is the standard deviation, C_1, C_2 are constant values used to avoid instability when $\left(2\mu_{I_{gray}}^2\mu_{I_{seg}}^2\right)$ approaches to zero.

Table.1. Performance speedup on various benchmarks using AMTIS compared to Otsu's Method in MATLAB.

Benchmark	#th	Running Time (in sec)		Speed-up
		AMTIS	Otsu	
Lena	2	0.0009	0.0141	14.88
	3	0.0009	0.0096	**10.21**
	4	0.0009	0.0108	11.67
	5	0.0009	0.0125	13.74
Cameraman	2	0.0004	0.0069	18.43
	3	0.0003	0.0095	28.88
	4	0.0003	0.0102	31.75
	5	0.0003	0.0109	33.39
Hunter	2	0.0005	0.0134	25.88
	3	0.0005	0.0151	30.68
	4	0.0005	0.0155	30.08
	5	0.0005	0.0163	32.72
Baboon	2	0.0033	0.0446	13.52
	3	0.0033	0.0449	13.61
	4	0.0033	0.0473	14.33
	5	0.0032	0.0470	14.69
Boat	2	0.0006	0.0145	25.41
	3	0.0005	0.0141	26.55
	4	0.0006	0.0166	26.40
	5	0.0005	0.0170	32.38
FingerPrint_1	2	0.0004	0.0073	17.51
	3	0.0003	0.0091	26.33
	4	0.0004	0.0095	27.00
	5	0.0003	0.0105	33.55
Blonde (Lady Zelda)	2	0.0005	0.0132	27.89
	3	0.0005	0.0139	28.18
	4	0.0005	0.0158	32.13
	5	0.0005	0.0164	**33.63**
Frozen Franz Joshef (Satellite Image)	2	0.1398	2.9871	21.37
	3	0.1356	2.9615	21.84
	4	0.1411	2.9811	21.13
	5	0.1420	2.9544	20.81

The FSIM [37], [23] calculates the similarity between two images: in this case, the original gray-scale image and the segmented image. In PSNR and SSIM, a higher value is considered for better performance of the thresholding method. The FSIM is then defined as:

$$FSIM = \frac{\sum_{x\in\Omega} S_L(x)PC_m(x)}{\sum_{x\in\Omega} PC_m(x)} \tag{16}$$

where $S_L(x) = S_{PC}(x)S_G(x)$

$$S_{PC}(x) = \frac{2PC_1(x)PC_2(x)+T_1}{PC_1^2(x)PC_2^2(x)+T_1}$$

$$S_G(x) = \frac{2G_1(x)G_2(x)+T_2}{G_1^2(x)G_2^2(x)+T_2} \tag{17}$$

The gradient magnitude of the image, G is given by:

$$G = \sqrt{G_x^2+G_y^2} \tag{18}$$

and PC is the phase congruence, expressed as

$$PC(x) = \frac{E(x)}{\varepsilon + \sum_n A_n(x)} \tag{19}$$

The local amplitude on the scale of n is $A_n(w)$ and $E(w)$ is taken to be the magnitude of the response vector in w on n. The term ε is a positive constant. For a better segmentation quality, a higher value of FSIM is desired.

The Table.1 shows the CPU computational time and our proposed algorithm's performance speed (AMTIS) to Otsu's method. A maximum speedup of 33.63x is observed for the Blonde benchmark and a minimum speedup of 10.21x for Lena, the most popular benchmark, respectively. Also, Table 2 presents the generated optimal threshold values along with PSNR, SSIM, and FSIM. We observed that the results obtained by our proposed algorithm AMTIS are comparable and better in many cases compared to the popular Otu's method. In a few benchmarks, AMTIS fails to compute optimal threshold values. This drawback is because we adopt a naïve approach in selecting the thresholds. One way to improve this is by devising an appropriate clustering technique.

The Fig.3 shows the outputs of the segmented image and the histogram showing thresholds obtained using AMTIS. The algorithm ensures that chosen thresholds are some valley point in the histogram. We see that human eyes can easily perceive images segmented using AMTIS.

5. CONCLUSIONS

We propose a heuristic approach for automatically segmenting an image to determine multilevel thresholds by sampling the histogram of a digital image. The algorithm first employs a gradient descent search to evaluate all valley points in the histogram of the input image. Secondly, the histogram is also partitioned into equal-sized regions to determine minimum frequency within each partition. This partitioning of a histogram is done to obtain candidate threshold values by eliminating multiple local valleys within a local region. It also ensures that candidate values are distributed uniformly in a histogram. Finally, in the third step, we emphasis valley points as optimal thresholds, based on a naive clustering approach. We find that such a naive approach is not very efficient for some benchmarks and required fine-tunning. One such improvement is to select the first

candidate threshold instead of taking the mean from the last cluster. As future work, appropriate clustering algorithms can be applied to select optimal threshold values. We demonstrated that our approach outperforms the popular Otsu's method in terms of CPU computational time. We observed a maximum speed-up of 35.58× and a minimum speed-up of 10.21× on popular image processing benchmarks. The results obtained by our proposed algorithm AMTIS are comparable and better in many cases in comparison to the popular Otu's method. We see that these images, segmented using AMTIS can be easily perceived by human eyes.

Table.2. Results on image segmentation using the multi-thresholds technique by AMTIS and Otsu's Method

Benchmark	#th	AMTIS Algorithm				Otsu's Algorithm			
		Thresholds Values	PSNR	SSIM	FSIM	Thresholds Values	PSNR	SSIM	FSIM
Lena	2	76 133	21.5534	0.8373	0.8730	93 162	18.3630	0.7767	0.8027
	3	55 133 159	17.7697	0.7298	0.7964	71 120 177	24.4511	0.8862	0.9078
	4	55 108 133 159	22.8782	0.8538	0.8827	56 100 145 193	21.9961	0.8093	0.8581
	5	55 76 108 149 159	23.6053	0.8608	0.8974	50 86 117 155 198	23.4954	0.8373	0.8776
Cameraman	2	56 120	23.5950	0.9038	0.8972	69 143	19.6920	0.8369	0.8373
	3	45 120 175	21.8643	0.8817	0.8706	59 121 157	23.7898	0.9054	0.9009
	4	34 81 154 191	18.7400	0.7841	0.8077	59 116 148 173	24.0384	0.8925	0.9010
	5	31 70 139 191 206	18.1175	0.8123	0.8195	45 97 135 162 196	24.3154	0.8620	0.8966
Hunter	2	82 132	23.5790	0.8357	0.8968	85 140	22.0429	0.8043	0.8680
	3	68 121 145	23.7282	0.8439	0.8976	69 111 153	26.9783	0.9019	0.9402
	4	58 101 156 190	20.5173	0.7523	0.8569	79 111 145 176	25.9988	0.8750	0.9407
	5	58 88 132 166 194	22.7267	0.7876	0.9046	71 110 141 161 185	25.0000	0.8507	0.9349
Baboon	2	56 97	27.3935	0.9439	0.9715	98 164	19.5858	0.7281	0.8603
	3	49 127 150	17.2279	0.7104	0.8362	73 123 178	22.7317	0.8448	0.9212
	4	42 80 127 134	22.2693	0.8518	0.9350	71 113 157 203	22.4076	0.8263	0.9327
	5	42 80 127 150 163	21.9339	0.8498	0.9411	51 87 122 160 204	22.9211	0.8525	0.9504
Boat	2	58 114	26.2578	0.9220	0.9378	92 154	17.7148	0.6612	0.8131
	3	42 106 138	25.1380	0.9152	0.9221	71 124 166	25.6688	0.8981	0.9314
	4	32 80 130 162	23.6828	0.8457	0.9071	60 111 145 178	23.4464	0.8055	0.9191
	5	26 69 106 162 186	18.3488	0.6885	0.8448	48 93 129 154 185	24.3153	0.8231	0.9363
FingerPrint_1	2	128 160	28.5674	0.9144	0.8949	155 195	24.4348	0.7809	0.7506
	3	96 152 181	22.4285	0.8510	0.8219	146 177 208	27.9377	0.8869	0.8651
	4	96 128 181 188	20.8015	0.7469	0.7208	122 154 182 210	26.0531	0.8790	0.8626
	5	90 113 137 160 181	29.7260	0.9481	0.9429	117 145 168 190 213	27.0431	0.9043	0.8983
Blonde (Lady Zelda)	2	59 91	28.8039	0.9078	0.9267	64 111	23.7733	0.8271	0.8465
	3	59 100 125	25.6890	0.8654	0.8844	53 92 125	26.3154	0.8758	0.8975
	4	43 70 91 113	29.6934	0.9087	0.9460	43 74 103 131	26.9770	0.8621	0.9104
	5	43 70 91 125 142	25.2996	0.8176	0.8812	39 67 92 114 136	27.2994	0.8549	0.9230
Frozen FranzJoshef (Satellite Image)	2	59 116	26.4262	0.9426	0.9792	65 157	19.6948	0.8620	0.9231
	3	59 108 125	28.3351	0.9559	0.9855	45 113 179	24.1935	0.9182	0.9655
	4	19 73 116 125	25.6461	0.9040	0.9736	33 89 146 196	23.5285	0.8889	0.9620
	5	19 73 116 154 206	24.5080	0.8781	0.9706	21 62 113 160 203	23.9781	0.8797	0.9687

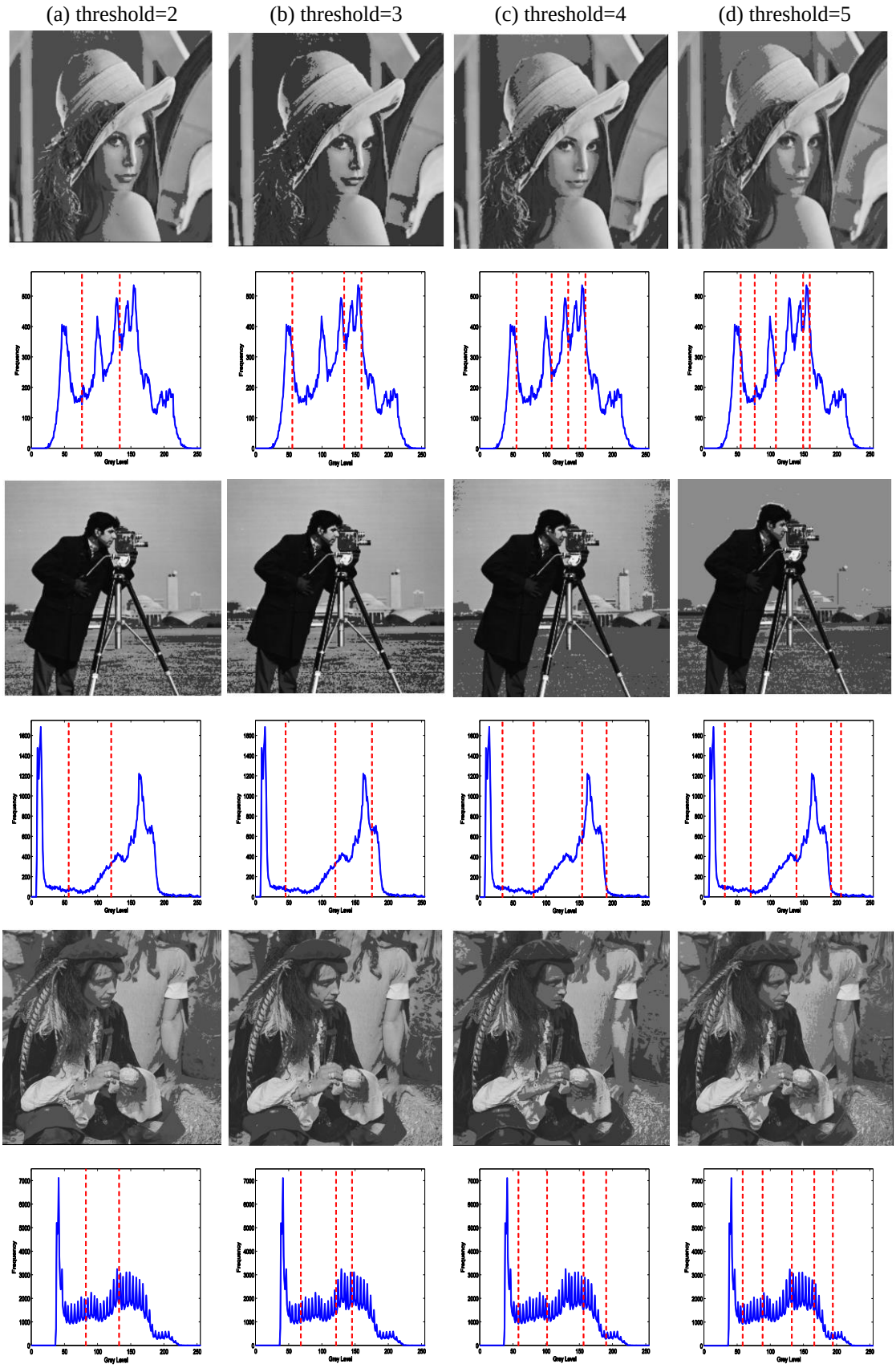

Fig.3. Results obtained using our approach on the benchmark Lena, Cameraman and Hunter

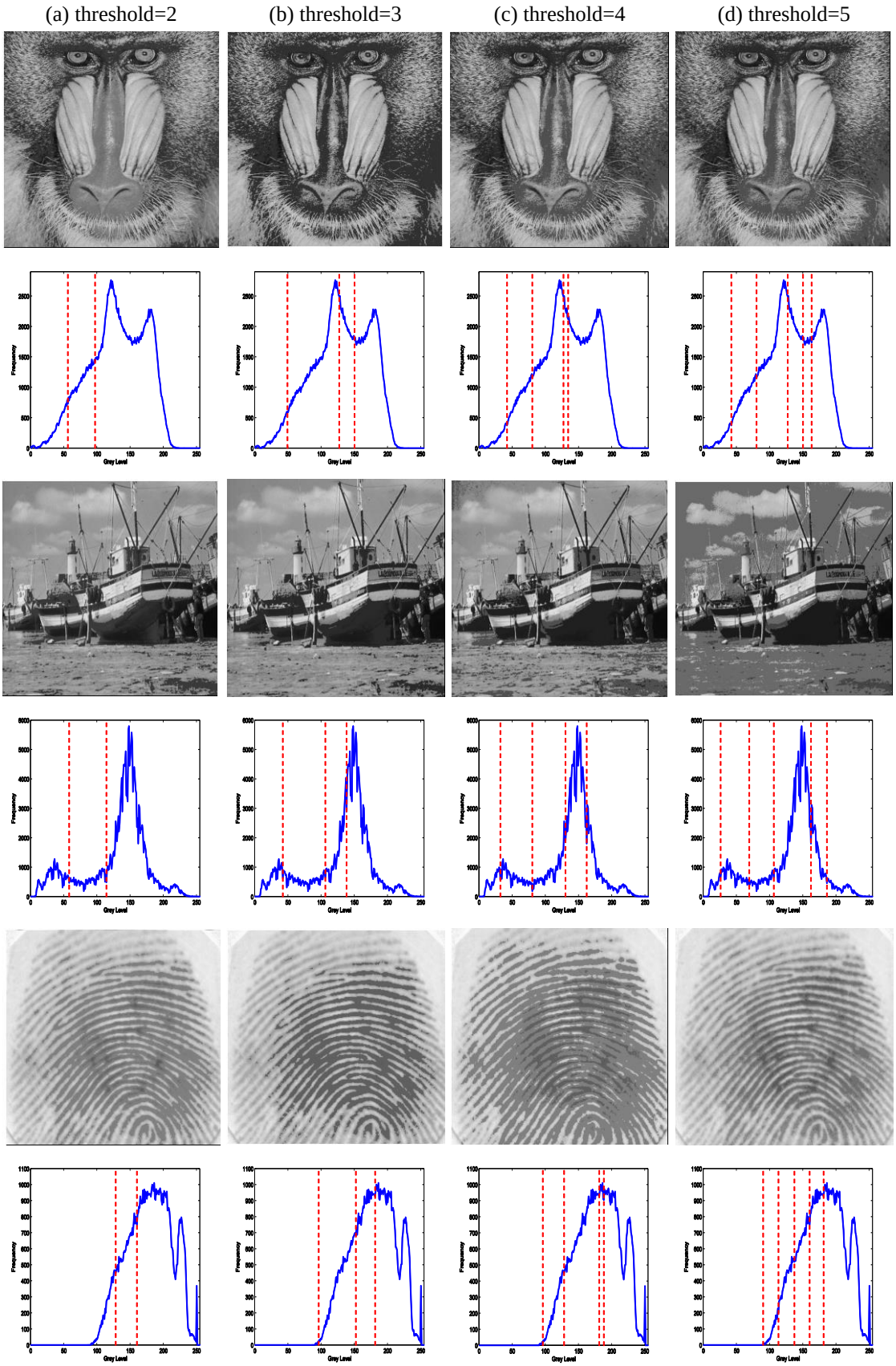

Fig.4. Result obtained using our approach on the benchmark Baboon, Boat and Fingerprint

REFERENCES

[1] B.S. Babu, S. Varadarajan and S. Swarnalatha, "Contrast Enhancement based Brain Tumour MRI Image Segmentation and Detection with Low Power Consumption", *I-Manager's Journal on Image Processing*, Vol. 3, No. 2, pp. 1-18, 2016.

[2] C. Li, R. Huang, Z. Ding, J.C. Gatenby, D.N. Metaxas and J.C. Gore, "A Level Set Method for Image Segmentation in the Presence of Intensity Inhomogeneities with Application to MRI", *IEEE Transactions on Image Processing*, Vol. 20, No. 7, pp. 2007-2016, 2011.

[3] F. Shi, Y. Fan, S. Tang, J.H. Gilmore, W. Lin and D. Shen, "Neonatal Brain Image Segmentation in Longitudinal MRI Studies", *Neuroimage*, Vol. 49, No. 1, pp. 391-400, 2010.

[4] R.P. Joseph, C.S. Singh and M. Manikandan, "Brain Tumor MRI Image Segmentation and Detection in Image Processing", *International Journal of Research in Engineering and Technology*, Vol. 3, No. 1, pp. 1-5, 2014.

[5] A. El-Sisi, "Design and Implementation Biometric Access Control System using Fingerprint for Restricted Area based on Gabor Filter", *The International Arab Journal of Information Technology*, Vol. 8, No. 4, pp. 355-363, 2011.

[6] B. Devereux, G. Amable and C.C. Posada, "An Efficient Image Segmentation Algorithm for Landscape Analysis", *International Journal of Applied Earth Observation and Geoinformation*, Vol. 6, No. 1, pp. 47-61, 2004.

[7] K. Yamaoka, T. Morimoto, H. Adachi, T. Koide and H.J. Mattausch, "Image Segmentation and Pattern Matching based FPGA/ASIC Implementation Architecture of Real-Time Object Tracking", *Proceedings of Asia and South Pacific Conference on Design Automation*, pp. 1-6, 2006.

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

[9] K.S. Fu and J. Mui, "A Survey on Image Segmentation", *Pattern Recognition*, Vol. 13, No. 1, pp. 3-16, 1981.

[10] W.X. Kang, Q.Q. Yang and R.P. Liang, "The Comparative Research on Image Segmentation Algorithms", *Proceedings of IEEE International Workshop on Education Technology and Computer Science*, pp. 703-707, 2009.

[11] J. Shi, J. Malik, "Normalized Cuts and Image Segmentation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 8, pp. 888-905, 2000.

[12] A. Bleau and L.J. Leon, "Watershed-Based Segmentation and Region Merging", *Computer Vision and Image Understanding*, Vol. 77, No. 3, pp. 317-370, 2000.

[13] S. Indira and A. Ramesh, "Image Segmentation using Artificial Neural Network and Genetic Algorithm: A Comparative Analysis", *Proceedings of IEEE International Conference on Process Automation, Control and Computing*, pp. 1-6, 2011.

[14] J. Wei and L. Chan, "An Image Segmentation Method based on Partial Differential Equation Models", *International Journal of Simulation–Systems, Science and Technology*, Vol. 17, No. 36, pp. 1-13, 2019.

[15] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol 9, No. 1, pp. 62-66, 1979.

[16] J.N. Kapur, P.K. Sahoo and A.K. Wong, "A New Method for Gray-Level Picture Thresholding using the Entropy of the Histogram", *Computer Vision, Graphics, and Image Processing*, Vol. 29, No. 3, pp. 273-285, 1985.

[17] P. Sathya and R. Kayalvizhi, "Optimal Multilevel Thresholding using Bacterial Foraging Algorithm", *Expert Systems with Applications*, Vol. 38, No. 12, pp. 15549-15564, 2011.

[18] S. Kullback, "*Information Theory and Statistics*", Courier Corporation, 1997.

[19] P.Y. Yin, "Multilevel Minimum Cross Entropy Threshold Selection based on Particle Swarm Optimization", *Applied mathematics and Computation*, Vol. 184, No. 2, pp. 503-513, 2007.

[20] M.H. Horng, "Multilevel Minimum Cross Entropy Threshold Selection based on the Honey Bee Mating Optimization", *Expert Systems with Applications*, Vol. 37, No. 6, pp. 4580-4592, 2010.

[21] R. Manikandan and M. Ramkumar, "Sequential Pattern Mining on Chemical Bonding Database in the Bioinformatics Field", *Proceedings of International Conference on AIP*, pp. 1-9, 2022.

[22] D. Oliva, E. Cuevas, G. Pajares, D. Zaldivar and V. Osuna, "A Multilevel Thresholding Algorithm using Electromagnetism Optimization", *Neurocomputing*, Vol. 139, pp. 357-381, 2014.

[23] A.K. Bhandari, V.K. Singh, A. Kumar and G.K. Singh, "Cuckoo Search Algorithm and Wind Driven Optimization based Study of Satellite Image Segmentation for Multilevel Thresholding using Kapurs Entropy", *Expert Systems with Applications*, Vol. 41, No. 7, pp. 3538-3560, 2014.

[24] A.K.M. Khairuzzaman and S. Chaudhury, "Multilevel Thresholding using Grey Wolf Optimizer for Image Segmentation", *Expert Systems with Applications*, Vol. 86, pp. 64-76, 2017.

[25] A.K. Bhandari, A. Kumar and G.K. Singh, "Modified Artificial Bee Colony based Computationally Efficient Multilevel Thresholding for Satellite Image Segmentation using Kapurs, Otsu and Tsallis Functions", *Expert Systems with Applications*, Vol. 42, pp. 1573-1601, 2015.

[26] H.F. Ng, "Automatic Thresholding for Defect Detection", *Pattern Recognition Letters*, Vol. 27, No. 14, pp. 1644-1649, 2006.

[27] P. Thangam, "*Digital Image Processing*", Charulatha, 2010.

[28] C. Poynton, "*Digital Video and HD: Algorithms and Interfaces*", Elsevier, 2012.

[29] A. Ismail and M. Marhaban, "A Simple Approach to Determine the Best Threshold Value for Automatic Image Thresholding", *Proceedings of IEEE International Conference on Signal and Image Processing Applications*, pp. 162-166, 2009.

[30] C.H. Li and C. Lee, "Minimum Cross Entropy Thresholding", *Pattern Recognition*, Vol. 26, No. 4, pp. 617-625, 1993.

[31] K. Tang, X. Yuan, T. Sun, J. Yang and S. Gao, "An Improved Scheme for Minimum Cross Entropy Threshold Selection based on Genetic Algorithm", *Knowledge-Based Systems*, Vol. 24, No. 8, pp. 1131-1138, 2011.

[32] D. Oliva, S. Hinojosa, V. Osuna-Enciso, E. Cuevas and G. Sanchez-Ante, "Image Segmentation by Minimum Cross

Entropy using Evolutionary Methods", *Soft Computing*, Vol. 23, No. 2, pp. 431-450, 2019.

[33] M.H. Horng and R.J. Liou, "Multilevel Minimum Cross Entropy Threshold Selection based on the Firefly Algorithm", *Expert Systems with Applications*, Vol. 38, No. 12, pp. 14805-14811, 2011.

[34] M. Maitra and A. Chatterjee, "A Hybrid Cooperative-Comprehensive Learning based PSO Algorithm for Image Segmentation using Multilevel Thresholding", *Expert Systems with Application*, Vol. 34, No. 2, pp. 1341-1350, 2008.

[35] S.T. Welstead, "*Fractal and Wavelet Image Compression Techniques*", SPIE Optical Engineering Press Bellingham, 1999.

[36] Z. Wang, A.C. Bovik, H.R. Sheikh and E.P. Simoncelli, "Image Quality Assessment: from Error Visibility to Structural Similarity", *IEEE Transactions on Image Processing*, Vol. 13, No. 4, pp. 600-612, 2004.

[37] L. Zhang, L. Zhang, X. Mou and D. Zhang, "FSIM: A Feature Similarity Index for Image Quality Assessment", *IEEE Transactions on Image Processing*, Vol. 20, No. 8, pp. 2378-2386, 2011.