QUANTUM INSPIRED GENETIC ALGORITHM FOR BI-LEVEL THRESHOLDING OF GRAY-SCALE IMAGES

Archana G. Pai, Krishna Mohan Buddhiraju, Surya S Durbha
Indian Institute of Technology, Bombay, India

KEY WORDS: Quantum Genetic Algorithm, Quantum Computing, Binary Threshold, Grey-Level Co-occurrence Matrix (GLCM)

ABSTRACT:

Thresholding is the primitive step in the process of image segmentation. Finding the optimal threshold for satellite images with reduced computation time and resources is still a challenging task. In this paper, we propose a Grey-Level Co-occurrence Matrix based Quantum Inspired Genetic Algorithm (QGA-GLCM) for bi-level thresholding of gray-scale images (natural and satellite). In this paper, QGA was used to find the optimal threshold. The results are compared with four different variants of Differential Evolution (DE) meta-heuristic algorithms, namely- DE-Otsu, DE-Kapur, DE-Tsali’s, DE-GLCM, and three different variants of QGA, namely- QGA-Otsu, QGA-Kapur, QGA-Tsali’s. Intensity value from image pixel is the only information used by Otsu, Tsali’s and Kapur for thresholding and are highly affected by noise. The main objective of this paper was a) To have a binary threshold for images corrupted with noise by bringing in spatial context b) To reduce the computational complexity and time for generating a threshold. Performance evaluators viz., CPU time, Peak Signal-to-Noise Ratio (PSNR), Mean Square Error (MSE), and Structural Similarity Index Measure (SSIM) were used for quantitative assessment of partitioned images. From this study we observed that our proposed technique, QGA-GLCM is a) very good at producing a diverse population b) ten times faster than its classical counterparts c) generates better threshold for images corrupted by noise. In general, the threshold values generated by QGA and its variants are better than its classical counterparts. The results clearly show that exploration and exploitation capability of QGA is superior to DE for all variants. QGA-GLCM can be an effective technique to generate thresholds both in terms of computational speed and time.

1. INTRODUCTION

Thresholding, is a widely explored and researched area with applications in all domains, like identifying faults and tumours, edge detection etc. Images acquired by satellites are at times corrupted by noise found in the sensor or transmission channel during acquisition and transmission. Segmenting satellite images is a process of grouping pixels into regions based on similarity resulting into a segmented image which labels each pixel and associates the pixel to its class. Thus formed homogeneous regions help to distinguish various objects present in the satellite images. Thresholding plays an important role in the separation of background from the foreground or an object from its surroundings. The number of thresholds used to segment an image can lead to bi-level or multi-level thresholding. The former classifies the image into two classes: foreground and background and the latter classifies the image into n+1 classes with ‘n’ different thresholds. Finding this single optimal threshold, or multiple optimal thresholds, is based on peaks and valleys in the histogram. A bi-modal histogram has two peaks and a valley. A threshold can be found around this valley. For images with multiple regions, the histogram is multi-modal, multiple peaks and valleys, generating multiple thresholds, one per valley for each region, and is called multi-level thresholding.

Thresholding, being a classical technique, relies on the information values contained in the pixel. If all the pixels present in the image contribute towards finding the threshold, then the technique is called global thresholding. If only some of the pixels are used to find the threshold, then the technique is called local thresholding. Global thresholding techniques are prone to noise and sensitive to illumination effects, shadows, etc., which is usually the case with satellite images. Due to their global nature, an exhaustive search needs to be carried out, which consumes computational resources exponentially with an increase in number of thresholds. To address the problem of exhaustive search, researchers used various combinations of meta-heuristic algorithms with different objective functions (Aziz et al., 2017; Bhandari et al., 2015a, 2015b, 2016; Mirjalili et al., 2016; Pare et al., 2017, 2018, 2020, 2021; Singh Gill et al., 2019; Upadhyay and Chhabra, 2020; Xing and Jia, 2019). 1D(dimension) thresholding techniques are fast, effective for real-world objects, and computationally less expensive but there are certain drawbacks with this approach: a) Two images with identical histogram leads to the same thresholds b) In the presence of noise and shadows, the performance of these histogram-based thresholding is poor. Researchers in the past addressed the problem of noisy images by including second order statistics like mean, median, and variance, as well as gradient magnitude and direction as additional information. This increases the algorithm complexity, time for processing and also the dimensionality of the pixel.

Recently, with the development in quantum computing, quantum inspired algorithms using principles of quantum mechanics are being used. Quantum-Inspired Genetic Algorithm (QGA) is one among them. It has good exploration and exploitation capability and hence needs very few iterations to reach a global optima(Xiong et al., 2018). Previously, QGA was used for bi-level and multi-level thresholding of gray/color images using different histogram-based objective functions (Dey et al., 2014, 2017; Hilali-Jaghdam et al., 2020; Sabeti et al., 2020).

In this work, we address the problem of finding threshold of noisy images using GLCM, which records the frequency distribution of gray level transitions, retaining the spatial relationship among pixels. The GLCM being symmetric in nature uses only upper triangular entries to find the threshold. Spatial context helps in reducing the noise and restricts the threshold search to be 1D. To the best of our knowledge, this is the first time a QGA is used along with edge information from the contrast feature of GLCM to find optimal threshold value of images. This combination of QGA and GLCM reduces the computational complexity both in terms of computational resources and time by exploiting the feature preservation capabilities of GLCM and the exploration skills of QGA, respectively. The rest of the paper is organized as follows: Section 2 presents the required background information about the various fitness functions, the algorithms used, the technologies used and the proposed methodology. Section 3 discusses in detail about experiments and results and finally, we conclude the paper with a conclusion.

2. BACKGROUND AND METHODOLOGY

This section deals with the required background and formulates the problem statement used in finding the threshold value of given image. Bi-level thresholding technique, partitions the pixels 'p' of the image into two regions based on intensity values(L) by selecting an appropriate threshold. Consider a gray-scale image ‘I’ of dimension M*N defined using ‘L’ grey levels.

$$p \rightarrow C_1 if \;\; 0 \le p \le th$$

$$\text{and}\, p \rightarrow C_2 \; if \;\; th+1 \le p \le L-1$$

where, $p \in I_{x,y}$ represents the pixel at location (x, y) of an image I, C1 and C2 represent the foreground and background classes respectively, and ‘th’ represent the threshold value. Section 2.1 explains in brief the various objective functions used. Section 2.2 introduces the basics of quantum computing. Section 2.3 briefly describes the various meta-heuristics algorithms used. Section 2.4 presents the proposed methodology.

2.1 Objective Functions

The efficiency of our proposed algorithm is compared using 3 different histogram based objective functions widely accepted in the literature. In the next section, we briefly discuss about each of them.

2.1.1 Tsali’s Entropy: Tsali’s entropy depends on two parameters: ‘th’ (threshold) and Tsali’s parameter ‘q’ (non-extensivity of a system). The apriori Tsali’s distribution for background and foreground is defined as (Portes de Albuquerque et al., 2004)

$$S_q^A(t) = \frac{1- \sum_{i=1}^{th}\left(\frac{p^i}{p^A}\right)^q}{q-1} \quad and \quad S_q^B(t) = \frac{1- \sum_{i=th+1}^{L}\left(\frac{p^i}{p^B}\right)^q}{q-1} \text{ where,}$$

$$p^A = \sum_{i=1}^{th} p^i \;\; and \;\; p^B = \sum_{i=th+1}^{L} p^i$$

The optimal threshold: $\tau^* = argmax\, S(t)$-------------------(1)

Where, $S(t) = S = S_q^A(t) + S_q^B(t) + (1-q)S_q^A(t) * S_q^B(t)$

Subject to: $|\, p^A + p^B| - 1 < S < 1 - \;|\, p^A + p^B|$

2.1.2 Kapur’s Entropy: It is a generalization of Shannon’s entropy. Kapur assumed the image to contain two probability distributions, corresponding to the object and the background, respectively. An optimal threshold value is one that maximises the sum of information content in the partitioned image (Kapur et al., 1985)

$$H_0 = -\sum_{i=0}^{th-1} \frac{p_i}{\omega_0} \ln\left(\frac{p_i}{\omega_0}\right) \text{ where } \omega_0 = \sum_{i=1}^{th-1} p_i$$

$$H_0 = -\sum_{i=th}^{L-1} \frac{p_i}{\omega_1} \ln\left(\frac{p_i}{\omega_1}\right) \text{ where } \omega_1 = \sum_{i=th}^{L-1} p_i$$

The optimal threshold $\tau^* = argmax\{H_0 + H_1\}$ ---------------(2)

2.1.3 Otsu Method: It is based on the criteria of between-class variance. It tries to maximize the distance between the classes while minimizing the variance within the class. This is expressed as a summation of sigma values for each of the classes (Otsu., 1979).

$$\sigma_0^2 = w_0(\mu_0 - \mu)^2 \text{ and } \sigma_1^2 = w_1(\mu_1 - \mu)^2$$

where μ denotes mean intensity of the image independent of threshold, background mean μ_0 and foreground mean μ_1 are given as,

$$\mu_0 = \sum_{i=1}^{th} \frac{i*p_i}{\omega_0}, \quad \omega_0 = \sum_{i=1}^{th} p_i, \text{ and}$$

$$\mu_1 = \sum_{i=th+1}^{L} \frac{i*p_i}{\omega_1}, \omega_1 = \sum_{i=th+1}^{L} p_i,$$

The optimal threshold $\tau^* = argmax\{\sigma_0^2 + \sigma_1^2\}$---------------(3)

2.1.4 Gray-scale co-occurrence matrix(GLCM): GLCM proposed by Haralick (Haralick et al., 1973) is a second order statistic which incorporates pair-wise coexistence of the gray-levels instead of intensity values of the pixel. GLCM considers the spatial relationship between a pair of pixels with parameters (d, θ). Each entry of GLCM at (i, j) location records the frequency of occurrence of intensities ‘i’ and ‘j’ between the pixel pairs separated by a constant distance ‘d’ and in the direction θ. For a given image with ‘L’ intensity levels, GLCM is a square matrix of dimension ‘L*L’. Each directional GLCM is sparse. The neighbouring pixel can be in one of the 4 standard directions (0º, 45º, 90º, 135º). Usually, the GLCM matrix is the average of these 4 standard directional matrices if directional features are not present in the image (Haralick et al., 1973).

$$GLCM = \frac{1}{4}\left[GLCM_{d,0^0} + GLCM_{d,45^0} + GLCM_{d,90^0} + \; GLCM_{d,135^0}\right]$$

Edge information available in the contrast feature of GLCM is used to calculate the value of ‘q’. It exploits the symmetrical property of the GLCM matrix and considers only the upper triangular entries of the GLCM matrix to calculate the threshold τ, reducing the computational complexity to half (Mokji & Abu Bakar, 2007).

$$\tau = \frac{1}{\eta} \sum_{m=0}^{L-1-q} \sum_{n=m+q}^{L-1} \left(\frac{m+n}{2}\right) GLCM(m,n)$$

Where, --------(4)

$$\eta = \sum_{m=0}^{L-1-q} \sum_{n=m+q}^{L-1} GLCM(m,n)$$

2.2 Quantum Computing

Quantum computing exploits quantum mechanical principles of superposition and entanglement to device quantum algorithms, and hence the algorithm generated is significantly faster (in terms of the reduced number of queries made, searching/evaluating all possible outcomes) than any classical algorithm when solving the same problem. Encouraged by this idea, currently, many computer science applications are developed in a hybrid model using quantum principles. Quantum computers are good at certain computational workloads where their classical counterparts are less efficient (Malossini et al., 2008). The smallest unit of information in quantum is a qubit. A bit in classical computing can be in one of the two states, viz., 0 or 1. A qubit, when in 0 or 1 state, is called the basis state and is said to be in superposition state, when in a linear combination of 0 and 1. Hence, the state of a general two-level qubit is given as $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ where, $|\alpha|^2 + |\beta|^2 = 1$. Each qubit is known as ‘ket’, which is a column vector $|\psi\rangle$ and its complex conjugate is known as ‘bra’, which is row vector $\langle\psi|$ named after the founder, Paul Dirac, representing the ‘bracket’ mathematically. An ‘n’ size quantum system is a collection of n qubits combined using a tensor product.

$$|1\rangle \otimes |0\rangle \ldots \otimes |0\rangle = |10 \ldots 0\rangle = |EDN\rangle$$

where ‘EDN’ represents the equivalent decimal number. Further information on qubit and its representation can be found in Nielsen (Nielsen and Chuang).

2.3 Meta-heuristic Algorithm

To avoid exhaustive search, population based meta-heuristic algorithms are used. This reduces the search to the size of the population and also spans the entire solution space, improving the solution iteratively. This session introduces us to the algorithms we used and compared viz., Quantum Genetic Algorithm (QGA) and Differential Evolution (DE).

2.3.1 Differential Evolution (DE): DE was proposed by Storn and Price. It is free from derivatives and is known as derivative-free optimization. It uses very few parameters and has faster convergence (Storn and Price, 1997). In the literature, many authors have used DE for bi-level, multi-level, and color thresholding of natural and satellite images using different objective functions. DE outperforms many of the conventional meta-heuristic algorithms, providing a good threshold (Bhandari et al., 2016; Pare et al., 2017; Upadhyay and Chhabra, 2020).

2.3.2 Quantum Genetic Algorithm (QGA): A Quantum genetic algorithm was introduced by Narayan and Moore (Narayanan and Moore, 1996) which combines the advantages of both classical and quantum mechanics. The Q-bit encoded chromosomes can represent a linear superposition of states probabilistically and can potentially map to a larger search space than other evolutionary algorithms (Zhang et al., 2014). The diversity of individuals in populations is greatly increased by the use of qubit instead of binary, or any other form of symbolic representation, which also helps to avoid premature convergence effectively. As a result, QGA proves to have higher efficiency and is faster.

2.4 Proposed Methodology

In this paper, we proposed QGA-GLCM which combines the exploration skills of QGA with spatial correlation among pixel intensities to find the optimal threshold. Our proposed quantum-inspired genetic algorithm uses GLCM as a fitness function to evaluate the individual chromosomes in the population. The proposed method reduces the computational time and aids in finding the optimal threshold. Figure 1 shows the flowchart of the proposed methodology. An explanation of the various steps in the proposed algorithm is given as follows:

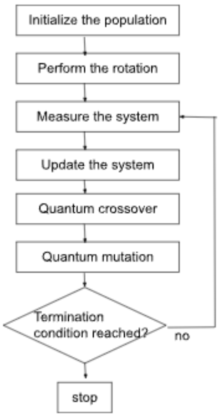


Figure 1: Flowchart of the proposed methodology

2.4.1 Initialize the population: QGA begins by initializing the population size (npop) and chromosome length(m), crossover probability (p_{cr}), mutation probability (p_{mut}), mutation rate for each gene (p_{mutg}) and the maximum number of generations (m_{gen}). Each gene in the chromosome is represented using a qubit. A combination of such m-qubits forms a chromosome representing each individual in the population. A set of npop chromosomes represents a population. The population is denoted by $Q = \{q_1, q_2, \dots, q_n\}$ where q_i represents the i$^{\text{th}}$ individual in the population. The population corresponding to generation ‘t’ is given by Q(t). Each qubit is initially set to $|0\rangle$ and later put into superposition of all states with equal probabilities using Hadamard gate.

$$q_i = \begin{pmatrix} \alpha_1| & \alpha_2| & \cdots & |\alpha_m \\ \beta_1| & \beta_2| & \cdots & |\beta_m \end{pmatrix}$$

2.4.2 Quantum rotation: The use of a rotation gate greatly reduces the algorithm search space becuase it is based on the table look-up. It also helps in improving the solution by advancing towards the global optimum. Each qubit is rotated by a phase angle θ, $\theta \in (0, \pi/2)$ producing a pair of amplitudes (α, β) that defines the state of j$^{\text{th}}$ qubit. The chromosome is initialized when all the qubits of chromosome are rotated. This completes the initialization process. This rotation can be achieved using a rotation matrix.

$$U(\theta) = \begin{bmatrix} cos\ \theta & -sin\ \theta \\ sin\ \theta & cos\ \theta \end{bmatrix}$$

2.4.3 Measure the system: This step converts the system from its superposition state back to binary. Let the qubit be represented by amplitudes (α, β) where $|\alpha|^2\ and\ |\beta|^2$ represents probability of getting zero and one respectively. Select a random number in the range (0, 1), and if the selected random number(r), $r < |\alpha|^2$ then the state of the corresponding qubit is set to 0 or else it is set to 1. This process is repeated for all the qubits of chromosome. This process measures the system of m-qubits probabilistically, leaving the system in one of its basis states to represent the chromosome. Each individual chromosome now represents an intensity value, which forms the initial set of intensity values. This process serves two purposes: it creates a classical population of the current quantum population to assist in evaluating the fitness function and it prevents the quantum system from collapsing to basis states (Lahoz-Beltra, 2016).

2.4.4 Update the system: On evaluating all chromosomes in population for the fitness, we obtain the best chromosome for the current iteration. The rotation gate is used as update operator of QGA to drive the individuals toward better solutions. They can strive the balance between exploration and exploitation. Rotation of a qubit by an arbitrary angle θ brings in randomness and helps in exploration of the search space. Rotation of a qubit is supported by look-up table (refer to Table 1) which is specific to a particular optimization problem. The angle $\delta\theta$ is used for updating the qubit, and the sign of $\delta\theta$ indicates the direction while its value indicates the angle of rotation. The speed of convergence of the solution directly depends on the value of $\delta\theta$ and needs to be selected carefully. Selection of a very high value for $\delta\theta$, results in a solutions diverging or converging to a local optimum. We update the qubits of the chromosome using rotation gate and it is represented as $Q(t+1) = U(t)Q(t)$ where $Q(t)$, $Q(t+1)$ represent the state of chromosomes in the population at ‘t’ and ‘t+1’ generations respectively. This is processed using rotation gate operator:

$$U(\theta) = \begin{bmatrix} cos\ \delta\theta & -sin\ \delta\theta \\ sin\ \delta\theta & cos\ \delta\theta \end{bmatrix}$$

C_i	B_i	$F(c_i) < F(B_i)$	$\delta\theta$
0	0	False	0
0	0	True	0
0	1	False	0
0	1	True	0.0785
1	0	False	0
1	0	True	-0.0785
1	1	False	0
1	1	True	0

Table 1: Look-up table for rotational angle($\delta\theta$)

2.4.5 Quantum crossover: This process resembles the classical crossover where the values are replaced by amplitudes of the qubits. The two individual chromosomes for mating are selected using the tournament selection process. Here, each chromosome undergoes crossover, if crossover probability (p_{cr}), is greater than the generated random number(r_{cr}), i.e. $r_{cr}< p_{cr}$. In this case, a random crossover point is selected and the amplitude values of all the qubits after this crossover point are interchanged, forming two new off-springs. Then a greedy selection is done between offspring and parents.

$$r_{cr} < p_{cr} \; then$$

$$q_1 = \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 & \alpha_5 \\ \beta_1 & \beta_2 & \beta_3 & \beta_4 & \beta_5 \end{bmatrix} \xrightarrow{crossover} q_1' \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3' & \alpha_4' & \alpha_5' \\ \beta_1 & \beta_2 & \beta_3' & \beta_4' & \beta_5' \end{bmatrix}$$

$$q_2 = \begin{bmatrix} \alpha_1' & \alpha_2' & \alpha_3' & \alpha_4' & \alpha_5' \\ \beta_1' & \beta_2' & \beta_3' & \beta_4' & \beta_5' \end{bmatrix} \xrightarrow{crossover} q_2' \begin{bmatrix} \alpha_1' & \alpha_2' & \alpha_3 & \alpha_4 & \alpha_5 \\ \beta_1' & \beta_2' & \beta_3 & \beta_4 & \beta_5 \end{bmatrix}$$

2.4.6 Quantum mutation: This process resembles the classical mutation, which is achieved by inverting the amplitudes of the qubit using an inversion gate. This process brings in diversity in the population. A chromosome mutates, if mutation probability (p_{mut}) is greater than the generated random number (r_{mut}) for the chromosome. i.e., $r_{mut} < p_{mut}$. Thus, if a chromosome is selected for mutation, then one or more genes in the chromosome mutate by swapping the amplitudes. A gene mutates if the mutation rate for the gene (p_{mutg}) is greater than random number generated for the gene (r_{mutg}). i.e., $r_{mutg} < p_{mutg}$. Let $(\alpha_{ik}, \beta_{ik})$ represent the amplitudes of kth qubit of ith chromosome then the mutation operation is given as:

$$r_{mutg} < p_{mutg} \text{ then } \binom{\alpha_{ik}}{\beta_{ik}} \rightarrow \binom{\beta_{ik}}{\alpha_{ik}}$$

2.4.7 Termination condition: Now measure the state of the population by evaluating the fitness value of each individual in the population. Obtain the best chromosome for the generation and optimal threshold value. Terminate the process, if current iteration is equal to m_{gen} else repeat the steps for measurement, to update the system, mutation, and crossover for m_{gen} number of times. The optimal threshold is the threshold value obtained at the end of m_{gen}.

3. EXPERIMENTS, RESULTS AND DISCUSSION

This session analyses the results of our proposed algorithm using various objective functions. The experimental set-up for this research work is described in Section 3.1. The obtained results are described in section 3.2.

3.1 Experimental Setup

The performance of the algorithm was evaluated for images from two groups: a) natural images consisting of an object and background from Berkeley Segmentation Dataset (Martin et al., 2001). b) satellite images of various regions collected from Google Earth, where the built-up areas (roads and buildings) form the object and the rest forms the background. We evaluated the performance of our algorithm for 10 images from each group. We compared the performance of algorithms quantitatively based on the CPU time required to execute the algorithm, Peak Signal-to-Noise Ratio (PSNR), Mean Square Error (MSE), and Structural Similarity Index (SSIM) measures. To evaluate the exploration capabilities of the algorithms, we used the mean and variance of fitnesses generated by the population in each generation. Finally, the partitioned images were validated for their quality of segmentation using K-Means algorithm with K=2.

3.2 Results

The behaviour of QGA-GLCM was analysed for efficiency and effectiveness and the results were compared with 4 different variants of DE algorithms, namely DE-Otsu, DE-Kapur, DE-Tsalis, DE-GLCM, and 3 different variants of QGA, namely QGA-Otsu, QGA-Kapur, QGA-Tsalis. The results obtained for three different images viz, an image from Berkeley Segmentation dataset (boat image) and two satellite images (corresponding to 'IITB campus' and 'Wadala Area' in suburban and Central Mumbai, collected from Google Earth) can be visualized in Figure 2. Figure 2(a) shows the original images. The segmented images, results of various algorithm, are shown in Figure 2(b-i). It was observed that QGA-GLCM has outperformed all other classical and quantum variants in case of boat, a noisy image. The threshold value generated by proposed method is 87 and by DE-GLCM is 151. Among the various segmented image in Figure 2(b-i), Figure 2(i) clearly shows the boat and its anchor. In case of satellite images, the partitioned image generated by QGA-GLCM is of better quality than its classical counterpart DE-GLCM, and is analogous to the one generated by QGA-Otsu. Numerically threshold values vary by a difference of 0 ± 10 between their respective classical and quantum variants for Otsu, Kapur and Tsali's objective functions and this difference varies from 0 ± 30 in case of GLCM based methods. Table 2 tabulates the results. The optimal threshold values generated by the proposed method and its variants are evaluated for 20 different images (10 from each group). A graph of optimal threshold value v/s images is plotted in Figure 3. From the figure, DE-Otsu and QGA-Otsu have almost the same threshold values for almost all images. For satellite images, the proposed algorithm considering the spatial context has almost the same threshold value as that of Otsu (both classical and quantum). For general images, it has outperformed Otsu sometimes, especially when the images are noisy. Threshold

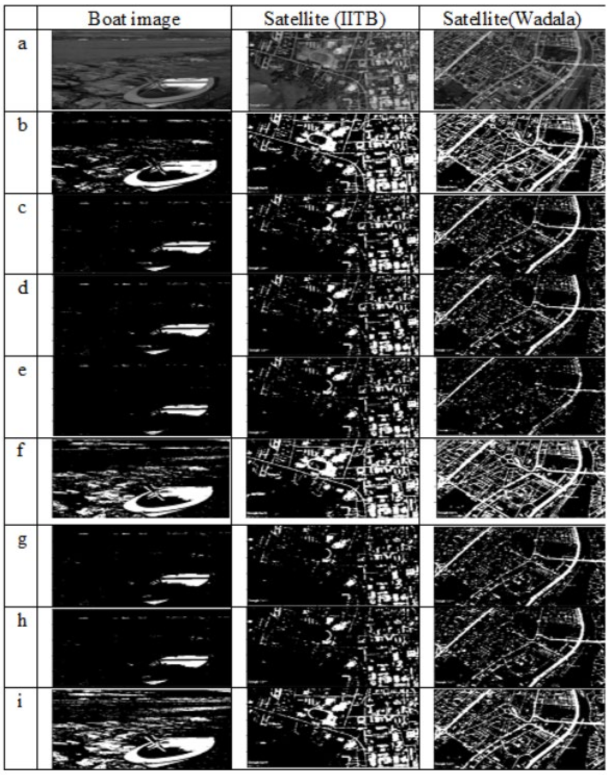
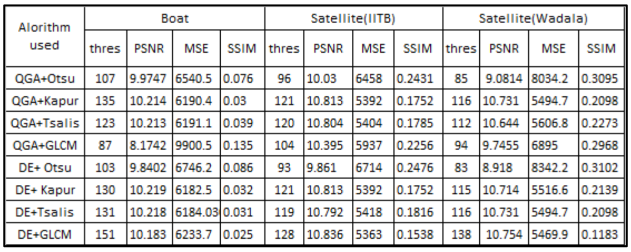


Figure 2: Segmented images using a) Original b) DE-OTSU c) DE-Kapur d) DE-Tsalis e) DE-GLCM f) QGA-OTSU g) QGA-Kapur h) QGA-Tsalis i) QGA-GLCM

values generated by GLCM as an objective function are always better than those generated by Kapur and Tsali’s as objective functions.

The graph of execution time taken v/s images by various algorithms to generate optimal threshold values can be visualized in Figure 4. On an average QGA-GLCM takes only 0.02seconds to generate threshold value for a given image, whereas DE-GLCM takes 2-3 seconds for the same. Due to huge difference of time between DE-GLCM and other variants, the time line corresponding to DE-GLCM is missing in the Figure 4. The QGA and its variants generate the threshold value for a given image in 0.002s-0.005s, which is 10 times faster than its classical variant which takes 0.02s-0.05s, proving the QGA and its variant’s computational efficiency.



Alorithm used	Boat				Satellite(IITB)				Satellite(Wadala)			
	thres	PSNR	MSE	SSIM	thres	PSNR	MSE	SSIM	thres	PSNR	MSE	SSIM
QGA+Otsu	107	9.9747	6540.5	0.076	96	10.03	6458	0.2431	85	9.0814	8034.2	0.3095
QGA+Kapur	135	10.214	6190.4	0.03	121	10.813	5392	0.1752	116	10.731	5494.7	0.2098
QGA+Tsalis	123	10.213	6191.1	0.039	120	10.804	5404	0.1785	112	10.644	5606.8	0.2273
QGA+GLCM	87	8.1742	9900.5	0.135	104	10.395	5937	0.2256	94	9.7455	6895	0.2968
DE+ Otsu	103	9.8402	6746.2	0.086	93	9.861	6714	0.2476	83	8.918	8342.2	0.3102
DE+ Kapur	130	10.219	6182.5	0.032	121	10.813	5392	0.1752	115	10.714	5516.6	0.2139
DE+Tsalis	131	10.218	6184.03	0.031	119	10.792	5418	0.1816	116	10.731	5494.7	0.2098
DE+GLCM	151	10.183	6233.7	0.025	128	10.836	5363	0.1538	138	10.754	5469.9	0.1183

Table 2: Comparison of Threshold, PSNR, MSE, and SSIM for different algorithm

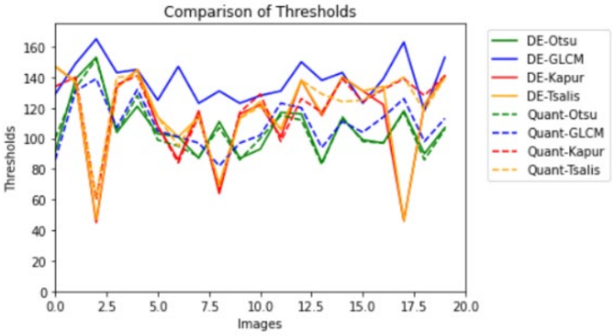


Figure 3: Comparing of thresholds of various algorithm for 20 different images

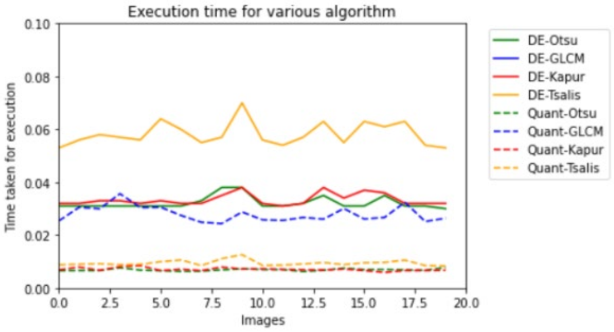


Figure 4: Comparing execution time of various algorithm for calculating thresholds for 20 different images

The average mean and average variance of the population were used in each generation to know more about the spread of the population. From the results, a) the variance of QGA and its variants are higher than DE and its variants b) average fitness value is lower than DE and its variants. This justifies the exploration capabilities of QGA. Table 3 tabulates the mean and variance of the fitness value generated in a generation. To compare the quality of the segmented image we used two-class K-Means classifier and found SSIM close to 1 for images where Otsu and GLCM were used as objective functions and above 0.7 for Tsalis and Kapur’s. Figure 5 shows a graph of SSIM values, quantifying the quality of the segmented image generated by various algorithms. From this figure it can be seen QGA-GLCM, DE-Otsu and QGA-Otsu have similar values showing high similarity of segmented image with the original image.

3.3 Conclusion

In this paper, we proposed a GLCM based QGA which uses edge information from contrast features. Including the contrast feature into GLCM has generated superior threshold values. Our experimental observations were as follows with respect to QGA and its variants: a) The average of the fitness values generated over various iterations has higher variance and low mean values, indicating that the solution space was less explored b) They are robust and have faster convergence (nearly 10 times faster). The proposed QGA-GLCM a) gives a better threshold for a noisy image without adding any additional information, and thus the segmented image is of superior quality, which is accounted for by higher SSIM value. b) The threshold values obtained are better than DE-GLCM and, in most cases, its equivalent to QGA-OTSU and DE-OTSU. c) The performance of the segmented image was compared with that of the two-class K-Means classifier using SSIM to obtain a value of almost 1. Notably, it can be concluded that the QGA-GLCM reduces overall computational time, generating a threshold value better than Otsu method for noisy images and equivalent to Otsu in other cases. Further, this research work can be extended for multilevel thresholding considering multiple classes.

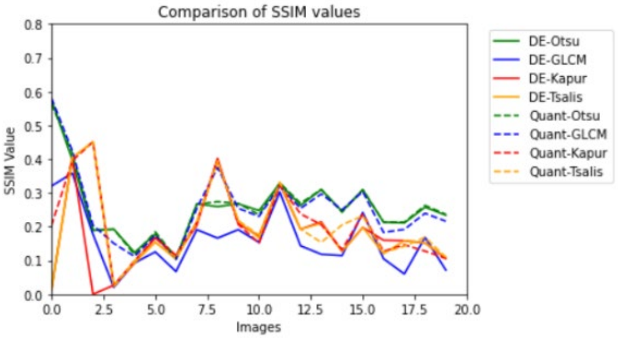
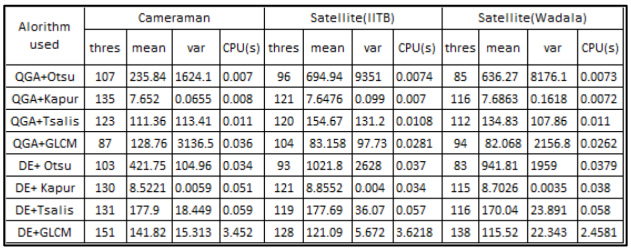


Figure 5: Comparing SSIM values of various algorithm for 20 different images



Alorithm used	Cameraman				Satellite(IITB)				Satellite(Wadala)			
	thres	mean	var	CPU(s)	thres	mean	var	CPU(s)	thres	mean	var	CPU(s)
QGA+Otsu	107	235.84	1624.1	0.007	96	694.94	9351	0.0074	85	636.27	8176.1	0.0073
QGA+Kapur	135	7.652	0.0655	0.008	121	7.6476	0.099	0.007	116	7.6863	0.1618	0.0072
QGA+Tsalis	123	111.36	113.41	0.011	120	154.67	131.2	0.0108	112	134.83	107.86	0.011
QGA+GLCM	87	128.76	3136.5	0.036	104	83.158	97.73	0.0281	94	82.068	2156.8	0.0262
DE+ Otsu	103	421.75	104.96	0.034	93	1021.8	2628	0.037	83	941.81	1959	0.0379
DE+ Kapur	130	8.5221	0.0059	0.051	121	8.8552	0.004	0.034	115	8.7026	0.0035	0.038
DE+Tsalis	131	177.9	18.449	0.059	119	177.69	36.07	0.057	116	170.04	23.891	0.058
DE+GLCM	151	141.82	15.313	3.452	128	121.09	5.672	3.6218	138	115.52	22.343	2.4581

Table 3: Comparison of Threshold, Mean, Variance, and CPU Time (in sec) for different algorithm

REFERENCES

Aziz, M. A. El, Ewees, A. A., & Hassanien, A. E. (2017). Whale Optimization Algorithm and Moth-Flame Optimization for multilevel thresholding image segmentation. *Expert Systems with Applications*, *83*, 242–256. https://doi.org/10.1016/j.eswa.2017.04.023

Bhandari, A. K., Kumar, A., Chaudhary, S., & Singh, G. K. (2016). A novel color image multilevel thresholding based segmentation using nature inspired optimization algorithms. *Expert Systems with Applications*, *63*, 112–

133. https://doi.org/10.1016/j.eswa.2016.06.044

Bhandari, A. K., Kumar, A., & Singh, G. K. (2015a). Modified artificial bee colony based computationally efficient multilevel thresholding for satellite image segmentation using Kapur's, Otsu and Tsallis functions. *Expert Systems with Applications*, *42*(3), 1573–1601. https://doi.org/10.1016/j.eswa.2014.09.049

Bhandari, A. K., Kumar, A., & Singh, G. K. (2015b). Tsallis entropy based multilevel thresholding for colored satellite image segmentation using evolutionary algorithms. *Expert Systems with Applications*, *42*(22), 8707–8730. https://doi.org/10.1016/j.eswa.2015.07.025

Dey, S., Bhattacharyya, S., & Maulik, U. (2017). Efficient quantum inspired meta-heuristics for multi-level true colour image thresholding. *Applied Soft Computing Journal*, *56*, 472–513. https://doi.org/10.1016/j.asoc.2016.04.024

Dey, S., Saha, I., Bhattacharyya, S., & Maulik, U. (2014). Multi-level thresholding using quantum inspired meta-heuristics. *Knowledge-Based Systems*, *67*, 373–400. https://doi.org/10.1016/j.knosys.2014.04.006

Haralick, R. M., Shanmugam, K., & Dinstein, I. (1973). Textural Features for Image Classification. *IEEE Transactions on Systems, Man, and Cybernetics*, *SMC-3*(6), 610–621. https://doi.org/10.1109/TSMC.1973.4309314

Hilali-Jaghdam, I., Ben Ishak, A., Abdel-Khalek, S., & Jamal, A. (2020). Quantum and classical genetic algorithms for multilevel segmentation of medical images: A comparative study. *Computer Communications*, *162*(June), 83–93. https://doi.org/10.1016/j.comcom.2020.08.010

Kapur, J. N., Sahoo, P. K., & Wong, A. K. C. (1985). A new method for gray-level picture thresholding using the entropy of the histogram. *Computer Vision, Graphics, & Image Processing*, *29*(3), 273–285. https://doi.org/10.1016/0734-189X(85)90125-2

Lahoz-Beltra, R. (2016). Quantum genetic algorithms for computer scientists. In *Computers* (Vol. 5, Issue 4). MDPI AG. https://doi.org/10.3390/computers5040024

Malossini, A., Blanzieri, E., & Calarco, T. (2008). Quantum genetic optimization. *IEEE Transactions on Evolutionary Computation*, *12*(2), 231–241. https://doi.org/10.1109/TEVC.2007.905006

Martin, D., Fowlkes, C., Tal, D., & Malik, J. (2001). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. *Proceedings of the IEEE International Conference on Computer Vision*, *2*, 416–423. https://doi.org/10.1109/ICCV.2001.937655

Mirjalili, S., Saremi, S., Mirjalili, S. M., & Coelho, L. D. S. (2016). Multi-objective grey wolf optimizer: A novel algorithm for multi-criterion optimization. *Expert Systems with Applications*, *47*, 106–119. https://doi.org/10.1016/j.eswa.2015.10.039

Mokji, M. M., & Abu Bakar, S. A. R. (2007). Adaptive Thresholding Based on Co-occurrence Matrix Edge Information. *Proceedings - 1st Asia International Conference on Modelling and Simulation: Asia Modelling Symposium 2007, AMS 2007*, *2*(8), 444–450. https://doi.org/10.1109/AMS.2007.8

Narayanan, A., & Moore, M. (1996). Quantum-inspired genetic algorithms. *Proceedings of the IEEE Conference on Evolutionary Computation*, 61–66. https://doi.org/10.1109/icec.1996.542334

Nielsen, M. A., & Chuang, I. L. (n.d.). *Quantum Computation and Quantum Information 10th Anniversary Edition*.

NOBUYUKI OTSU. (1979). A Tlreshold Selection Method from Gray-Level Histograms. *IEEE Transaction on Systems, Man and Cybernetics*, *20*(1), 62–66.

Pare, S., Bhandari, A. K., Kumar, A., & Singh, G. K. (2017). An optimal color image multilevel thresholding technique using grey-level co-occurrence matrix. *Expert Systems with Applications*, *87*, 335–362. https://doi.org/10.1016/j.eswa.2017.06.021

Pare, S., Kumar, A., & Singh, G. K. (2018). Color multilevel thresholding using gray-level co-occurrence matrix and differential evolution algorithm. *Proceedings of the 2017 IEEE International Conference on Communication and Signal Processing, ICCSP 2017*, *2018-Janua*, 96–100. https://doi.org/10.1109/ICCSP.2017.8286622

Pare, S., Kumar, A., Singh, G. K., & Bajaj, V. (2020). Image Segmentation Using Multilevel Thresholding: A Research Review. *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, *44*(1), 1–29. https://doi.org/10.1007/s40998-019-00251-1

Pare, S., Mittal, H., Sajid, M., Bansal, J. C., Saxena, A., Jan, T., Pedrycz, W., & Prasad, M. (2021). Remote sensing imagery segmentation: A hybrid approach. *Remote Sensing*, *13*(22). https://doi.org/10.3390/rs13224604

Portes de Albuquerque, M., Esquef, I. A., Gesualdi Mello, A. R., & Portes de Albuquerque, M. (2004). Image thresholding using Tsallis entropy. *Pattern Recognition Letters*, *25*(9), 1059–1065. https://doi.org/10.1016/j.patrec.2004.03.003

Sabeti, M., Karimi, L., Honarvar, N., & Boostani, R. (2020). Quantumized Genetic Algorithm for Segmentation and Optimization Tasks. *Biomedical Engineering: Applications, Basis and Communications*, *32*, 2050022. https://doi.org/10.4015/S1016237220500222

Singh Gill, H., Singh Khehra, B., Singh, A., & Kaur, L. (2019). Teaching-learning-based optimization algorithm to minimize cross entropy for Selecting multilevel threshold values. *Egyptian Informatics Journal*, *20*(1), 11–25. https://doi.org/10.1016/j.eij.2018.03.006

Storn, R., & Price, K. (1997). Differential Evolution-A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. In *Journal of Global Optimization* (Vol. 11). Kluwer Academic Publishers.

Upadhyay, P., & Chhabra, J. K. (2020). Kapur's entropy based optimal multilevel image segmentation using Crow Search Algorithm. *Applied Soft Computing*, *97*, 105522. https://doi.org/10.1016/j.asoc.2019.105522

Xing, Z., & Jia, H. (2019). Multilevel Color Image Segmentation Based on GLCM and Improved Salp Swarm Algorithm. *IEEE Access*, *7*, 37672–37690. https://doi.org/10.1109/ACCESS.2019.2904511

Xiong, H., Wu, Z., Fan, H., Li, G., & Jiang, G. (2018). Quantum rotation gate in quantum-inspired evolutionary algorithm: A review, analysis and comparison study. *Swarm and Evolutionary Computation*, *42*(February), 43–57. https://doi.org/10.1016/j.swevo.2018.02.020

Zhang, J., Li, H., Tang, Z., Lu, Q., Zheng, X., & Zhou, J. (2014). An improved quantum-inspired genetic algorithm for image multilevel thresholding segmentation. *Mathematical Problems in Engineering*, *2014*. https://doi.org/10.1155/2014/295402