MEDICAL IMAGE ENHANCEMENT USING EVOLUTIONARY ALGORITHM

An Industrial Oriented Mini Project (CS704PC)

Submitted

in partial fulfillment of the requirements for

the award of the degree of

Bachelor of Technology

in

Computer Science and Engineering

by

Ms. S. LAXMI PRANITHA (18261A05A8)

Under the Guidance of

Mr. P. SATYA SHEKAR VARMA
(Assistant Professor)

Dept. of Computer Science and Engineering

MAHATMA GANDHI INSTITUTE OF TECHNOLOGY

HYDERABAD - 500075

2021-2022

MAHATMA GANDHI INSTITUTE OF TECHNOLOGY

(Affiliated to Jawaharlal Nehru Technological University Hyderabad)
Gandipet, Hyderabad - 500075, Telangana.

Certificate

This is to certify that the project entitled **Medical Image Enhancement Using Evolutionary Algorithm** being submitted by **Ms. S. Laxmi Pranitha** bearing roll no:**18261A05A8** in partial fulfilment for the award of **Bachelor of Technology** in **Computer Science and Engineering** to **Jawaharlal Nehru Technological University Hyderabad** is a record of bonafide work carried out by her under our guidance and supervision.

Guide	Head of the Department
Mr. P. Satya Shekar Varma	**Dr. C.R.K. Reddy**
Assistant Professor	Professor

External Examiner

Declaration

This is to certify that the work reported in this project titled “**Medical Image Enhancement Using Evolutionary Algorithm”** is a record of work done by me in the Department of Computer Science and Engineering, Mahatma Gandhi Institute of Technology, Hyderabad.

No part of the work is copied from books/journals/internet and wherever the portion is taken, the same has been duly referred in the text. This report is based on the work done entirely by me and not copied from any other source.

The results embodied in this project have not been submitted to any other university or Institute for the award of any degree or diploma.

S. LAXMI PRANITHA
H.T. No: 18261A05A8

Acknowledgement

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of people who made it possible because success is the abstract of hard work and perseverance, but steadfast of all is encouraging guidance. So, I acknowledge all those whose guidance and encouragement served as a beacon light and crowned my efforts with success.

I would like to express my sincere thanks to my guide **Mr. P. Satya Shekar Varma**, **Assistant Professor**, Department of CSE, MGIT, for his constant guidance, encouragement and moral support throughout the project.

I am extremely thankful to **Dr. A. Nagesh, Professor, Dr. V. Subba Ramaiah, Sr. Assistant Professor and A. Ratnaraju, Assistant Professor**, Department of CSE, Industrial Oriented Mini Project Coordinators, MGIT, for their encouragement and support throughout the project.

I convey my heartfelt thanks to **Dr. C.R.K Reddy**, **Professor & HOD**, Department of Computer Science and Engineering, MGIT, for all the timely support and valuable suggestions during the period of project.

I am also thankful to **Dr. K. Jaya Sankar, Principal MGIT**, for providing the work facilities in the college.

Finally, I would like to thank all the faculty and staff of CSE Department who helped me directly or indirectly, for completing this project.

S.LAXMI PRANITHA
H.T. No:18261A05A8

Table of contents

List of figures

List of tables

Abstract

In Magnetic Resonance Imaging (MRI), the poor quality images may not provide the sufficient information for the visual interpretation of the affected locations of human body. So, to improve the image visions and to provide computational support, a novel adaptive image enhancement technique has been proposed named as Medical image enhancement using Evolutionary algorithm. The proposed framework includes Evolutionary algorithm, histogram sub-division and modified probability density function (PDF). A novel approach of subdivision is applied to the histogram using the exposure threshold and optimal threshold for preserving the brightness and reducing the information loss. To make the proposed technique more adaptive, the threshold parameters are optimized by utilizing the concept of Evolutionary algorithm, guided by the proposed multi-objective fitness function. Then, the PDF of each sub-histogram is modified to enhance the image quality. The experimental results show that, the proposed technique performs superior over other existing enhancement techniques.

1. Introduction

Image enhancement is one of the significant image processing techniques for improving the image quality. It is also required to make the image good enough for further processing and interpretation. It is used as a pre-processing technique in various medical imaging systems such as Computed Tomography (CT), X-ray, MRI. In general, medical images are of very poor quality, low contrast.

Histogram equalization (HE) is one of the traditionally used technique for improving the image contrast by re-scaling the dynamic range and histogram distribution of the image. But, annoying artefacts and noise amplification are noticed in histogram equalized images. Texture region based histogram equalization is used in this algorithm for minimizing the artefacts and enhance the contrast. Many algorithms have been developed for image quality enhancement, but most of the methods are not suitable for enhancement of poorly illuminated medical images, as they result artefacts in the enhanced images. Few techniques enhance the brightness of the image but not able to distinguish the object and back ground part effectively. The main challenges observed in medical images are entropy maximization, PSNR, contrast improvement, reducing artefacts, brightness preservation and maintaining the structure and feature similarity. Genetic algorithm is used to make the enhancement technique more adaptive. The main contribution of this work is presented below.

i. A novel approach of image sub-division has been proposed.

ii. Modification of Probability density function (PDF) of each sub-

iii. A novel multi-objective fitness function is proposed in this paper which is defined by taking functions like edge contents, texture parameters, PSNR, AMBE.

iv. The threshold parameters are searched automatically without human intervention which makes our proposed GAAHE algorithm more adaptive.

1.1 Problem Definition

Medical images are very complex in nature due to presence of several overlapped objects. These are affected by artifacts and geometric distortions. So for better diagnosis, artifacts are to be avoided. The object may not be observed in low contrast images. So, understanding of such images is very difficult and the diagnosis process becomes crippled which results wrong measurements during diagnosis.

1.2 Existing System

HE improves image contrast by re-scaling the dynamic range and histogram distribution of the image. However HE is not reliable as some unnecessary artefacts and noise amplification are noticed in histogram equalized images. HE also impacts the overall brightness of the image. To preserve the brightness and to improve the contrast of the images, different enhancement techniques like, brightness preserving bi-histogram equalization (BBHE) , dualistic sub-image histogram equalization (DSIHE) minimum mean brightness error bi-histogram equalization (MMBEBHE) have been developed. But, the above discussed methods are not explaining the mechanism of controlling the enhancement rate. A new technique for enhancement of low light images named as recursively separated exposure based sub-image histogram equalization (RS-ESIHE) .Each technique has its own limitations and drawbacks.

1.3 Proposed System

The proposed system is implemented on Spyder using python . The main aim of this project is to reduce information loss, preserving brightness, reducing noise.The information loss in existing techniques can be reduced by using image sub-division approach. Based on exposure threshold and adaptive optimal threshold parameter of each sub-histogram the image brightness can be preserved using a evolutionary algorithm. To improve the evolutionary algorithms usability an objective function can be proposed. The parameters like edge contents, contrast, PSNR, AMBE, entropy, energy are used to define a multi-objective fitness function. Equal weightage of 25% is given to each objective function. The threshold parameters are searched automatically therefore making the genetic algorithm more adaptive.

1.4 Requirements Specification

1.4.1 Software Requirements

The following details are the software requirements for Medical Image Enhancement using Evolutionary Algorithm.

Operating System	: Windows 10, Linux, Mac OS
Programming Language	: Python
Packages	: Numpy, Opencv, Matplotlib, Skimage, Genetic Algorithm
IDE	: Pycharm

1.4.2 Hardware Requirements

The following details are the hardware requirements for Medical Image Enhancement using Evolutionary Algorithm.

RAM	: 4GB
Processor	: Intel i3 or higher
Hard Disk	: 20GB.

1.5 Software Tools Used

1.5.1 Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Python is simple and easy to learn. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed. Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy and a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. The Proposed System works on python 3.5 and above.

1.5.2 Spyder

Spyder is an open-source cross-platform IDE. The Python Spyder IDE is written completely in Python. It is designed by scientists and is exclusively for scientists, data analysts, and engineers. It is also known as the Scientific Python Development IDE and has a huge set of remarkable features. It offers a unique combination of the advanced editing, analysis, debugging, and profiling functionality of a comprehensive development tool with the data exploration, interactive execution, deep inspection, and beautiful visualization capabilities of a scientific package. Beyond its many built-in features, its abilities can be extended even further via its plugin system and API. Furthermore, Spyder can also be used as a PyQt5 extension library, allowing developers to build upon its functionality and embed its components, such as the interactive console, in their own PyQt software.

2. Literature Survey

“Genetic algorithm based adaptive histogram equalization (GAAHE) technique for medical image enhancement” is developed by Upendra kumar Acharya, Sandeep Kumar *et.al.* It was published in 2021. They have used Genetic Algorithm on medical images. It has drawbacks like less entropy[1].

“Particle swarm optimized texture based histogram equalization (PSOTHE) for MRI brain image enhancement” is developed by Upendra kumar Acharya, Sandeep Kumar *et.al.* It was published in 2020. They have used Particle Swarm Optimization Algorithm on medical images. It has drawbacks like it is easy to fall into local optimum in high-dimensional space and has a low convergence rate in the iterative process[2].

“A novel reformed histogram equalization based medical image contrast enhancement using krill herd optimization” is developed by Pankaj Kandhway, Ashish Kumar, Anurag Singh *et.al.* It was published in 2020. They have used Krill Herd Optimization Algorithm on medical images. It has drawbacks like it is slower than other methods[3].

“Brightness preserving Bi-histogram equalization using edge pixels information” is developed by Md. Moniruzzaman , Md. Shafuzzaman , Md. Foisal Hossain *et.al.* It was published in 2014. They have used Histogram Equalization technique on medical images. It has drawbacks like it Produces some unwarranted artifacts in the final image[4].

Table 2.1 shows the literature survey of Medical Image Enhancement Using Evolutionary Algorithm. It contains name of year, author, title of proposed work, method, drawbacks.

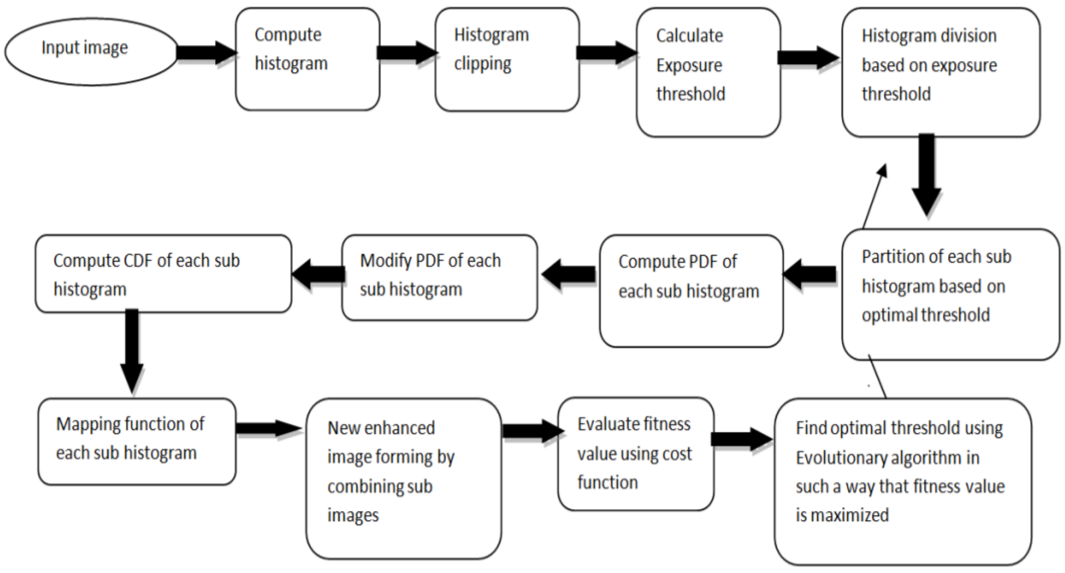
Table 2.1: Literature survey of Medical Image Enhancement Using Evolutionary Algorithm

S.No	Year	Author	Title	Method	Drawbacks
1.	2021	Upendra kumar Acharya, Sandeep Kumar	Genetic algorithm based adaptive histogram equalization (GAAHE) technique for medical image enhancement	Genetic Algorithm	Entropy is decreased.
2.	2020	Upendra kumar Acharya, Sandeep Kumar	Particle swarm optimized texture based histogram equalization (PSOTHE) for MRI brain image enhancement	Particle Swarm Optimization (PSO) Algorithm	Easy to fall into local optimum in high-dimensional space and has a low convergence rate in the iterative process
3.	2020	Pankaj Kandhway. Ashish Kumar Bhandari, Anurag Singh	A novel reformed histogram equalization based medical image contrast enhancement using krill herd optimization	Krill herd optimization	Slower than other methods
4.	2014	Md. Moniruzzaman, Md. Shafuzzaman, Md. Foisal Hossain	Brightness preserving Bi-histogram equalization using edge pixels information	Histogram equalization technique for image contrast enhancement which preserves brightness	Produces some unwarranted artefacts in the final image.

3. Methodology

3.1 System Architecture

In proposed system, initially the image dataset's is taken as input. Histogram is computed for image and it is clipped. Based o exposure threshold histogram is sub divided. By using fitness function optimal threshold values are obtained, based on these values again histograms are subdivided. PDF is calculated and it is modified and then CDF is calculated. Transfer functions are calculated. Using 4 objective functions final fitness value is calculated such that fitness value is maximized by giving equal weightage i.e.25% to each objective function. Finally image after enhancement using genetic algorithm is obtained. It has better entropy than GAAHE technique which uses only 3 functions.



Input image
Compute histogram
Histogram clipping
Calculate Exposure threshold
Histogram division based on exposure threshold
Partition of each sub histogram based on optimal threshold
Compute PDF of each sub histogram
Modify PDF of each sub histogram
Compute CDF of each sub histogram
Mapping function of each sub histogram
New enhanced image forming by combining sub images
Evaluate fitness value using cost function
Find optimal threshold using Evolutionary algorithm in such a way that fitness value is maximized

Figure 3.1 System Architecture of proposed method

Steps in proposed method

Step-1: Input image histogram is computed.

Step-2: Clipping threshold is calculated by taking the average of mean and median of intensity value. A new histogram is formed by clipping the original histogram using clipping threshold.

Step-3: Exposure threshold (Et) of the original histogram is calculated. It is used to divide the histogram into two sub-histograms . The sub histograms represent the low exposure and high exposure part of the image.

Step-4: Now each sub-histogram is subdivided into two sub-histograms using optimal threshold parameters Etl and Etu.

Step-5: The PDF in each sub-histogram is modified using the current PDF, sum of the PDF value.

Step-6: The thresholds used to form the sub-histograms are optimized in such a way that, the fitness function is to be maximized with respect to the iterations. As the medical images are more complex, so genetic algorithm is used to optimize the thresholds.

Step-7: CDF of each sub-histogram is calculated using modified PDF. Then the mapping function of each sub-histogram is computed using the CDF of each sub-histogram.

Step-8: Finally, a new enhanced image is obtained by combining each equalized sub-image.

Step-9: Then fitness value of each image is computed. This method is continued until the termination criterion is satisfied or the fitness curve becomes convergence.

Calculating Clipping Threshold

Clipping threshold (Cavg) is computed to clip the histogram. The main purpose of clipping the histogram is to avoid over enhancement and control the enhancement rate. This clipping threshold is calculated by taking the average of mean and median of image intensity values.

$$C_{median} = \text{median}\,(h(k))$$

$$C_{mean} = \text{mean}(h(k))$$

$$C_{avg} = \frac{C_{median} + C_{mean}}{2}$$

Exposure Threshold Calculation

The exposure threshold is calculated to divide the histogram into two sub-histograms which result two sub-images. These subimages are under exposed sub-image and over exposed sub-image. The normalized value of exposure lies in between 0–1. Exposure value and Exposure threshold of image is calculated as

$$Exposure = \frac{1}{L}\frac{\sum_{k=1}^{L} h(k)k}{\sum_{k=1}^{L} h(k)}$$

$$E_t = L \times (1 \text{ - Exposure})$$

L - max gray level

h(k) - histogram of the image.

Proposed Threshold selection

To preserve the brightness and reducing the information loss, the method of histogram sub-division is applied in some algorithms. To make the algorithm more adaptive, it is required to select the threshold parameter automatically depending upon few parameters. Taking this into mind, the thresholds Etl and Etu used to divide the sub-histogram are to be selected automatically using Genetic algorithm as per the proposed multi-objective fitness function which make the algorithm more adaptive.

Calculation of PDF of each sub-image

PDF of each sub image is calculated as

$$P_{l1}(k) = \frac{h_{cl}(k)}{N_{l1}}, for\ 0 \le k \le E_{tl}$$

$$P_{l2}(k) = \frac{h_{cl}(k)}{N_{l2}}, for\ E_{tl} + 1 \le k \le E_t$$

$$P_{u1}(k) = \frac{h_{cl}(k)}{N_{u1}}, for\ E_t + 1 \le k \le E_{tu}$$

$$P_{u2}(k) = \frac{h_{cl}(k)}{N_{u2}}, for\ E_{tu} + 1 \le k \le L - 1$$

Nl1, Nl2, Nu1, Nu2 are numbers of pixels present in sub-images I1, I2, I3, I4.

Proposed modification in PDF of each sub-image

PDF of each sub image is modified using previous PDF and summation of PDF of sub images.

For sub-image-1(Extremely low exposure sub-image), the modified PDF is,

$$P_{l1m} = \left(\frac{P_{l1}}{\sum_{k=0}^{k=E_{tl}} (P_{l1}(k)) + P_{l1}} \right)$$

For sub-image-2 (low exposure sub-image), the modified PDF is

$$P_{l2m} = \left(\frac{P_{l2}}{\sum_{k=E_{tl}+1}^{k=E_t} (P_{l2}(k)) + P_{l2}} \right)$$

For sub-image-3 (high exposure sub-image), the modified PDF is

$$P_{u1m} = \left(\frac{P_{u1}}{\sum_{k=E_t+1}^{k=E_{tu}} (P_{u1}(k)) + P_{u1}} \right)$$

For sub-image-4 (Extremely high exposure sub-image), the modified PDF is

$$P_{u2m} = \left(\frac{P_{u2}}{\sum_{k=E_{tu}+1}^{k=L-1} (P_{u2}(k)) + P_{u2}} \right)$$

Calculation of CDF for each sub-image

Cumulative Density Function (CDF) of each sub image is calculated as

$$C_{l1}(k) = \sum_{k=0}^{k=E_{tl}} P_{l1m}(k)$$

$$C_{l2}(k) = \sum_{k=E_{tl}+1}^{k=E_t} P_{l2m}(k)$$

$$C_{u1}(k) = \sum_{k=E_t+1}^{k=E_{tu}} P_{u1m}(k)$$

$$C_{u2}(k) = \sum_{k=E_{tu}+1}^{k=L-1} P_{u2m}(k)$$

Mapping function for each sub-image

The transfer function for each sub-histogram is computed

$$T_{l1} = (E_{tl}) \times C_{l1}$$

$$T_{l2} = (E_{tl}+1) + (E_t - (E_{tl}+1)) \times C_{l2}$$

$$T_{u1} = (E_t+1) + (E_{tu} - (E_t+1)) \times C_{u1}$$

$$T_{u2} = (E_{tu}+1) + (L - (E_{tu}+1)) \times C_{u2}$$

Proposed fitness function

Four objective functions are considered to calculate the fitness function and maximize it. They are -

1. Gray-level co-occurrence matrix (GLCM)

The texture of image is characterized using GLCM

2. Sobel Edge Detection Method

Edges present in image are detected using this method

3. Peak Signal to Noise Ratio (PSNR)

PSNR used to measure anti noise performance of algorithms

4. Absolute Mean Brightness Error (AMBE)

AMBE helps to find out quality of image in terms of brightness preservation.

3.2 Genetic Algorithm

Genetic algorithms are used in this project to calculate optimal sub exposure thresholds. Genetic algorithms are a class of algorithms whose search methods are based on natural evolution. They are probabilistic algorithms. Genetic algorithms are domain independent.

Five phases are considered in a genetic algorithm.

1. Initial population
2. Fitness function
3. Selection
4. Crossover
5. Mutation

Initial Population

The process begins with a set of individuals which is called a Population. Each individual is a solution to the problem you want to solve. An individual is characterized by a set of parameters (variables) known as Genes. Genes are joined into a string to form a Chromosome (solution).In a genetic algorithm, the set of genes of an individual is represented using a string, in terms of an alphabet. Usually, binary values are used (string of 1s and 0s). It can be said that we encode the genes in a chromosome.

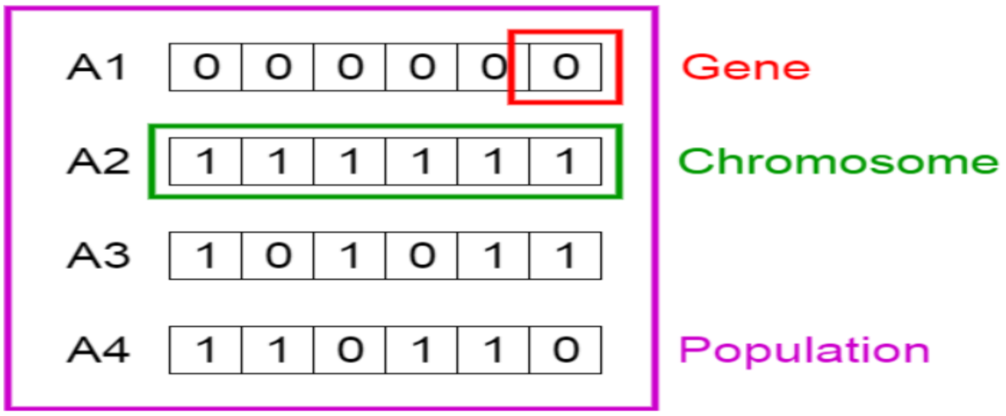


Figure 3.2 Population, Chromosomes and Genes

Fitness Function

The fitness function determines how fit an individual is (the ability of an individual to compete with other individuals). It gives a fitness score to each individual. The probability that an individual will be selected for reproduction is based on its fitness score.

Selection

The idea of selection phase is to select the fittest individuals and let them pass their genes to the next generation. Two pairs of individuals (**parents**) are selected based on their fitness scores. Individuals with high fitness have more chance to be selected for reproduction.

Crossover

Crossover is the most significant phase in a genetic algorithm. For each pair of parents to be mated, a crossover point is chosen at random from within the genes. Offspring are created by exchanging the genes of parents among themselves until the crossover point is reached. The new offspring are added to the population.

Mutation

In certain new offspring formed, some of their genes can be subjected to a mutation with a low random probability. This implies that some of the bits in the bit string can be flipped.

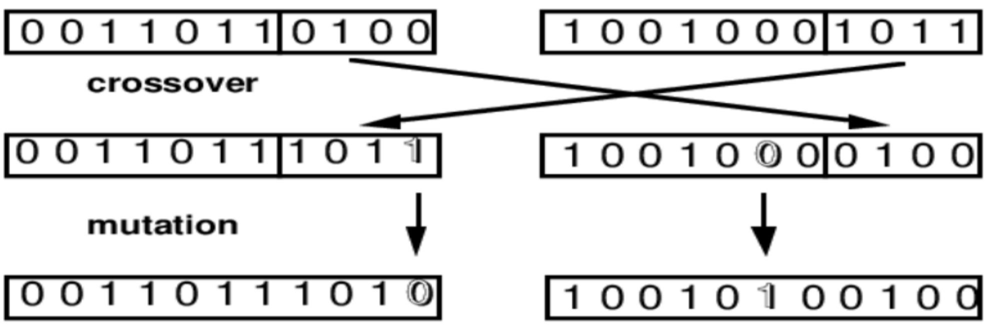


Figure 3.3 Crossover and Mutation in genetic algorithm

Genetic algorithm is a class of evolutionary computing technique, used to generate high quality solutions during searching process. The algorithm begins with a set of solutions (represented by chromosomes) called population. Solutions from one population are taken and are used to form a new population. Below steps have been followed to execute the proposed method,

Step-1: Generate the random population of n chromosomes.

Step-2: Evaluate the fitness of each chromosome in the population.

Step-3: Create a new population by repeating the following operations like selection, crossover and mutation. Then new offspring is placed in the new population.

Step-4: Use the new generated population for a further run of the algorithm

Step-5: If the end condition is satisfied, stop and return the best solution in current population.

Step-6: If the termination condition is not satisfied then go to Step 2.

3.3 UML Diagrams

UML is an acronym that stands for Unified Modeling Language. Simply put, UML is a modern approach to modeling and documenting software. In fact, it's one of the most popular business process modeling techniques. It is based on diagrammatic representations of software components. As the old proverb says: "a picture is worth a thousand words". By using visual representations, we are able to better understand possible flaws or errors in software or business processes.

3.3.1 Use Case Diagram

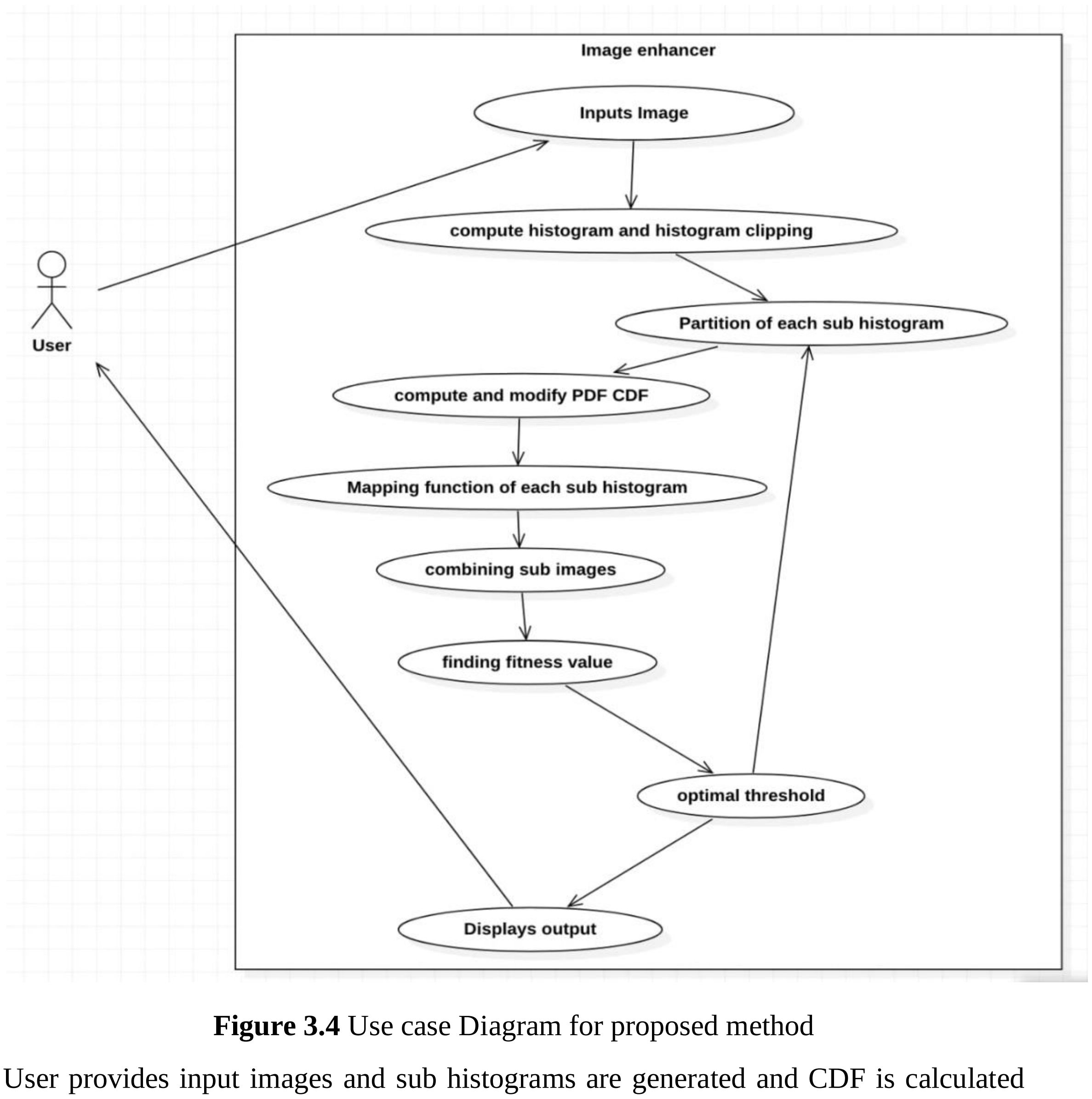


Image enhancer
Inputs Image
compute histogram and histogram clipping
Partition of each sub histogram
compute and modify PDF CDF
Mapping function of each sub histogram
combining sub images
finding fitness value
optimal threshold
Displays output
User

Figure 3.4 Use case Diagram for proposed method

User provides input images and sub histograms are generated and CDF is calculated and sub images are combined, iterations are done until fitness value is maximized. Then output image is displayed.

Sequence Diagram:

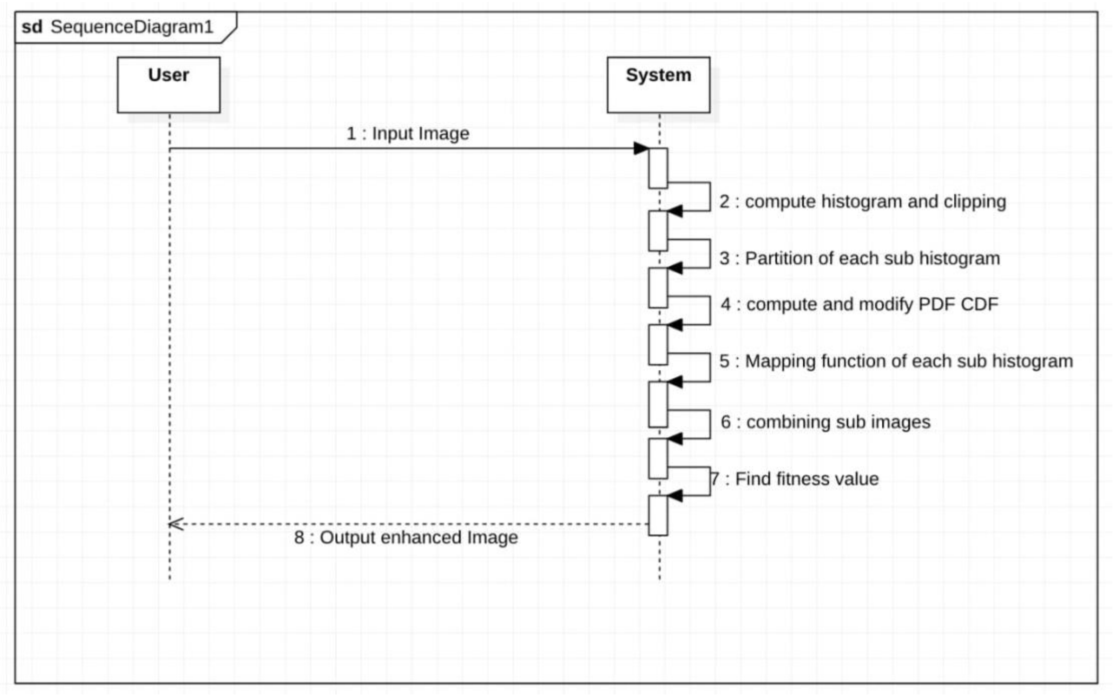

sd SequenceDiagram1
User
System
1 : Input Image
2 : compute histogram and clipping
3 : Partition of each sub histogram
4 : compute and modify PDF CDF
5 : Mapping function of each sub histogram
6 : combining sub images
7 : Find fitness value
8 : Output enhanced Image

Figure 3.5 Sequence Diagram for proposed method

In this system user provides the input image to the system. Then system computes histogram and clipping and it partitions each sub histogram and PDF is modified. CDF is computed and mapping function of each sub histogram is done. All sub images are combined and fitness value is calculated. This process continues until fitness value is maximized. At last output enhanced image is displayed.

4. Implementation and Results

4.1 Fitness Function

The medical images are of low contrast and noisy. Information loss is also more in the enhanced medical images. It is also very difficult to preserve the image feature after enhancement. So in this project, a novel fitness function has been proposed by taking the parameters like contrast, information contents (entropy), PSNR, edge contents and energy. To calculate the optimal sub exposure thresholds, genetic algorithms are used with the following fitness function as the objective to maximize. Multi-objective function is used with each objective given equal weightage.

The proposed fitness function is represented as,

cf = 0.25 × cf1 + 0.25 × cf2 + 0.25 × cf3+0.25 x cf4

4.1.1 First objective of the fitness function

Usually medical images are of inferior quality due to low contrast and less information. Thus it is paramount to preserve the information present in the image while increasing the contrast of the image. It is done by using the first objective for preserving entropy of the image while comparing it with the contrast.

$$cf1 = \log\left(\left(I_{contrast} \times \exp\left(I_{entropy}\right)\right)/I_{energy}\right)$$

To calculate contrast, entropy and energy the following equations are required

$$I_{contrast} = |i-j|^2 \times \mathrm{GL(i,\ j)}$$

$$I_{entropy} = -\sum_{i=0}^{n-1} pd_i log_2 pd_i$$

$$I_{energy} = \sum_{i,\,j} GL(i,\ j)^2$$

Here GL is the gray level co-occurrence matrix which is used for calculating texture of an image. The following equations are used for calculating GL matrix. Both energy and contrast have been computed using Gray-level co-occurrence matrix (GLCM). The texture of an image is characterized by the GLCM functions. Each element (i, j) in the resultant GLCM is represented by the sum of the number of times that the pixel with the value i occurred in the specified spatial relationship to a pixel with value j in the input image.

$$GL(i,j) = \frac{g(i,\ j)}{\sum_{i=0}^{H-1} \sum_{j=0}^{V-1} g(i,\ j)}$$

$$g(i,\ j) = \sum_{k=0}^{H-1} \sum_{l=0}^{V-1} \partial(k,\ l)$$

$$\partial(k,\ l) = \begin{cases} 1,\ If\ I_e(k,\ l) = i\ and\ I_e(k,\ l+1) = j \\ \quad or\ I_e(k,\ l) = i\ and\ I_e(k+1,\ l) = j \\ \qquad 0,\ otherswise \end{cases}$$

Variable H - number of pixels present in horizontal direction of image

Variable V - number of pixels present in vertical direction of image

Ienergy - the energy which indicates the homogeneity in the image. It indicates the pixel pair repetition within the image.

Icontrast - Local variations of gray levels are measured. If the neighboring gray level differences is more, then the image is of higher contrast.

Ientropy - The entropy of the image. It reveals the average information contents within the image. If the intensity distribution is uniform then it can be said that the histogram is equalized and contain higher entropy.

n - maximum intensity value of the image

pdi - PDF at gray level i.

4.1.2 Second objective of the fitness function

Second objective function is taken as edge contents of the image. It is represented by cf2. Sobel edge detection has been used to detect the edges in the images. So the second fitness function is evaluated by taking these parameters into consideration. Isobel represents the Sobel edge image, E(Isobel) is the sum of the intensity of the edge pixels and n edges represents the number of edge pixels of the resulting image and w x x represents the size of the image.

$$cf2 = log(\log(E(I_{sobel}))) \times \frac{n_{edges}}{w \times x}$$

4.1.3 Third objective of the fitness function

Most techniques while equalizing contrast levels introduce some level of noise to the image. This noise may interfere with the diagnosis of the disease. Thus to keep the noise level to a minimum use introduce PSNR (Peak Signal Noise Ratio) parameter as the third objective for the fitness function. Higher the PSNR value indicates better the anti-noise performance of the algorithm.

$$\mathrm{cf3} = PSNR = 10log_{10}\left(\frac{255^2}{MSE}\right)$$

Whereas the MSE is calculated as

$$MSE = \frac{1}{r \times c}\sum_{i=1}^{r}\sum_{j=1}^{c}(I_o(i,j) - I_e(i,j))^2$$

4.1.4 Fourth objective of the fitness function

Finally to preserve the brightness level in the image, AMBE (Absolute mean brightness error) is used. Lower is the AMBE, more is the brightness preservation in the enhanced image. The difference between mean brightness of input and output image is known as AMBE. This parameter helps to find out the quality of the image in terms of brightness preservation.

cf4=-AMBE(I,Y)

$$\mathrm{AMBE}\ (I,\ Y) = |\mathrm{E}(I) - E(Y)|$$

4.2 How to run project

1) Install Spyder

2) Install different packages like geneticalgorithm, image-similarity measures, matplotlib,numpy, opencv-python, pillow, scikit-image .

3) Import these required packages into files.

4) Write different python programs like for cdf, pdf, main ... different python files must be created.

5) In main.py file take input image

6) Run main.py file

4.3 Results

4.3.1 Original Histogram of Input Image1

An example medical image of a chest xray (size 138 x 197) is taken as follows

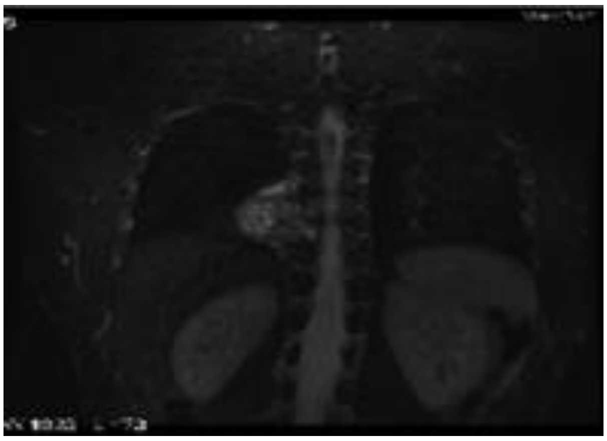


Figure 4.1.1 Original Image1

Histogram for above image is as follows

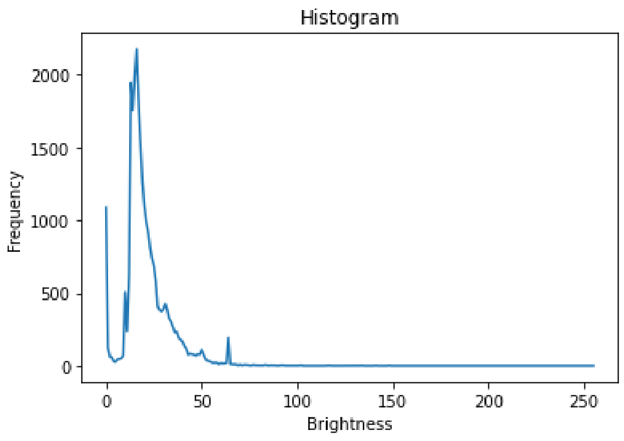


Figure 4.1.2 Histogram of Original Image1

The entropy of the image is 5.0447

It is the image given as input. It is the image of chest xray.

Exposure threshold of above image is at 219.Histogram is plotted by taking brightness on x-axis and Frequency on y-axis.

4.3.2 Previous methods of Histogram Equalization

Histogram equalization to input image1 results in

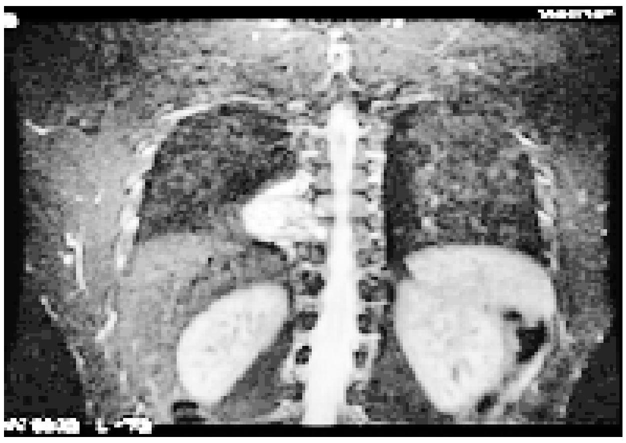


Figure 4.1.3 Equalized Image after HE

Histogram for above image is as follows

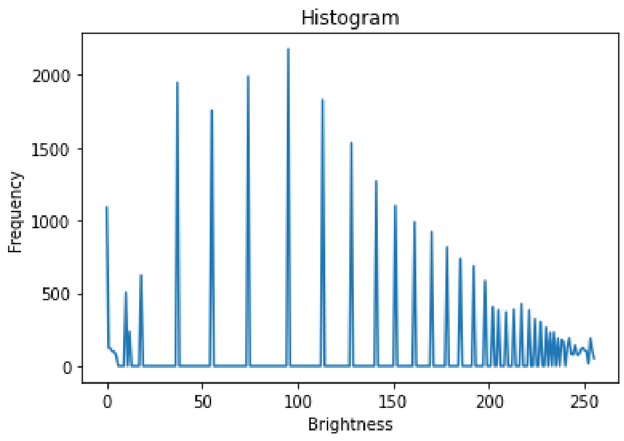


Figure 4.1.4 Histogram after HE

Measures of Entropy, AMBE, PSNR, SSIM for assessing the quality of the image are used. The values of these measures for histogram are as follows

Entropy: 4.7013

AMBE: 106.7375

PSNR: 4.8316

SSIM: 0.4244

4.3.3 GAAHE technique for Input Image1

Image after applying GAAHE(Genetic algorithm based adaptive histogram equalization) technique results in

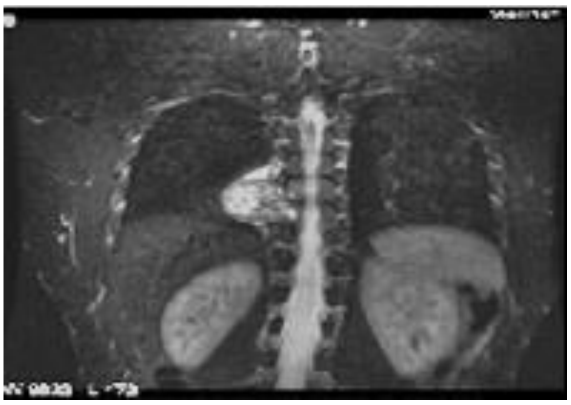


Figure 4.1.5 Equalized Image after GAAHE

Objective Function: -16.9818

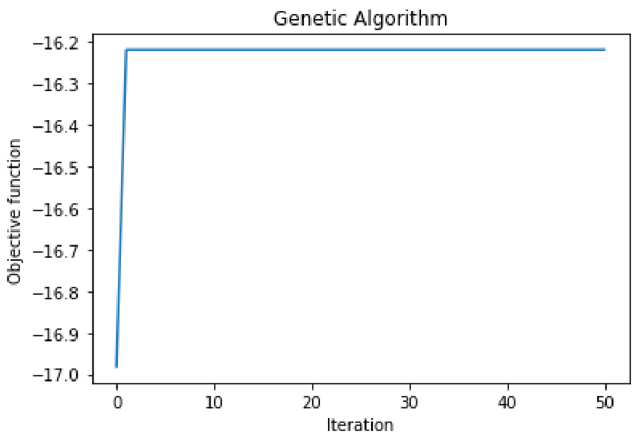


Figure 4.1.6 GA Histogram for Image1 before changing parameters

Histogram for above image is as follows

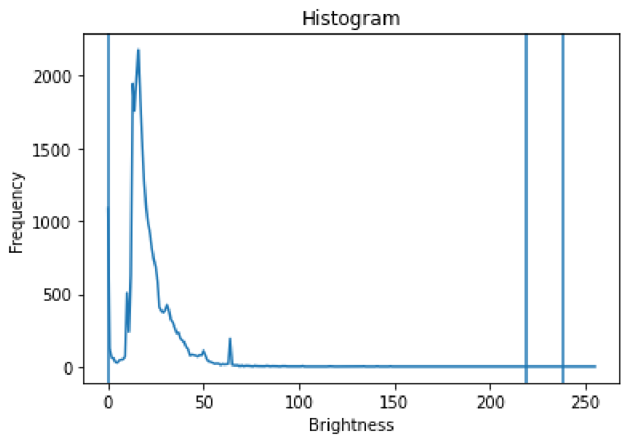


Figure 4.1.7 Histogram after GAAHE

The values of the image assessment measures are as follows for GAAHE,

Entropy: 5.03246

AMBE: 50.6127

PSNR: 12.9222

SSIM: 0.6632

4.3.4 Proposed Method

The input image1 after equalization with the proposed method is as follows

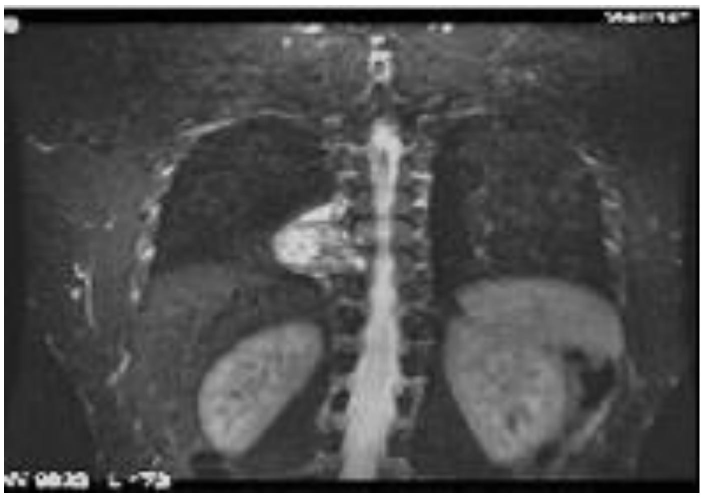


Figure 4.1.8 Equalized Image after applying proposed method

Objective Function: -17.3651

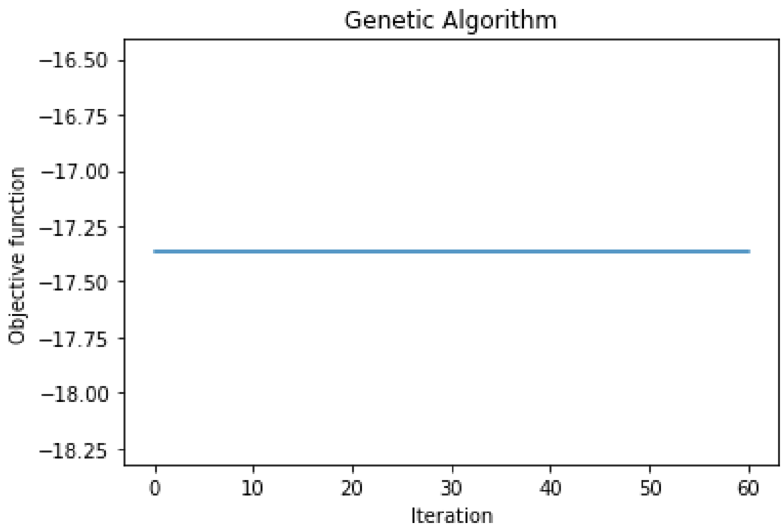


Figure 4.1.9 GA Histogram for Image1 after changing parameters

Histogram for above image is

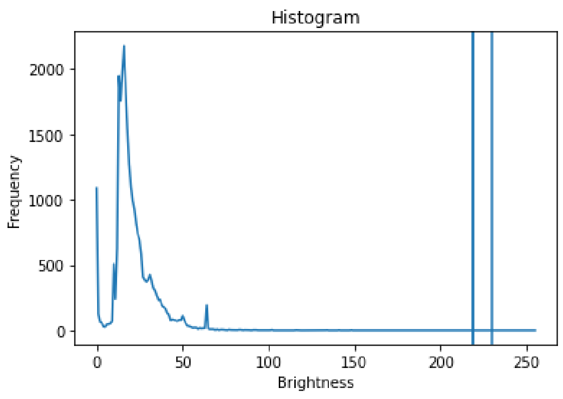


Figure 4.1.10 Histogram after proposed method

The values of the image assessment measures are as follows for proposed method,

Entropy: 5.0330

AMBE: 52.3786

PSNR: 12.7256

SSIM: 0.6497

Enhanced image is obtained using proposed method. To compare proposed method with other method Histogram equalization and GAAHE methods are taken into consideration so that histograms and images can be compared and also function value and entropy can also be compared between these methods. Parameters are changed in proposed method when compared to GAAHE method. And also in proposed method four objective functions are considered whereas in GAAHE method only three objective functions are considered. There is change in population_size , max_num_iterations, parents_proportion parameters between proposed method and GAAHE method.

4.3.5 Original Histogram of Input Image2

An example medical image of a neck xray (size 147 x 185) is taken as follows

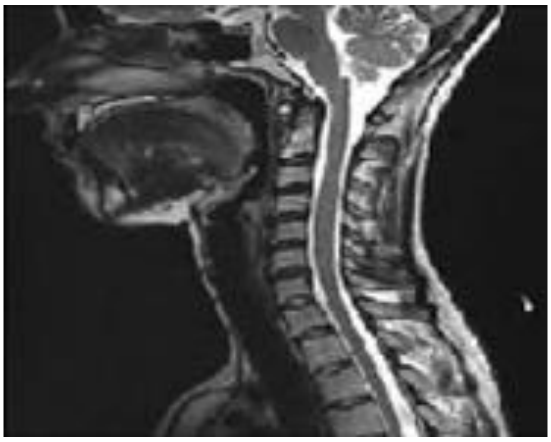


Figure 4.2.1 Original Image2

Histogram for above image is as follows

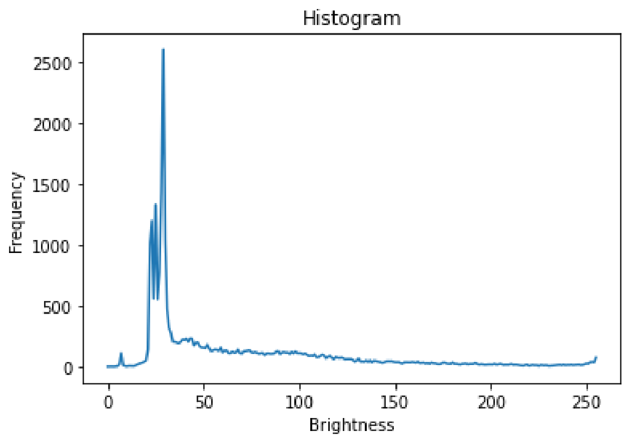


Figure 4.2.2 Histogram of Original Image2

The entropy of the image is 6.631

This is the second image that is given as input. It is the image of neck xray.
Exposure threshold of above image is at 190.Histogram is plotted by taking brightness on x-axis and Frequency on y-axis.

4.3.6 Previous methods of Histogram Equalization

Histogram equalization to input image2 results in

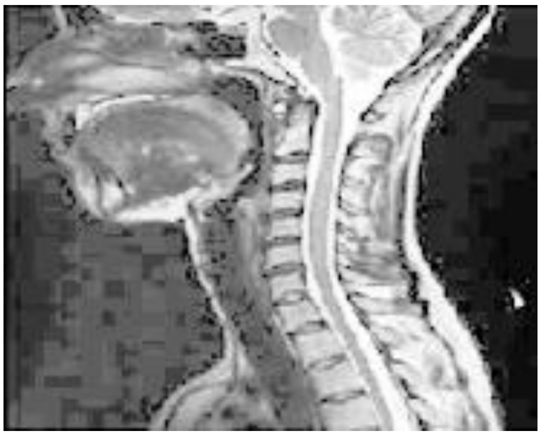


Figure 4.2.3 Equalized Image after HE

Histogram for above image is as follows

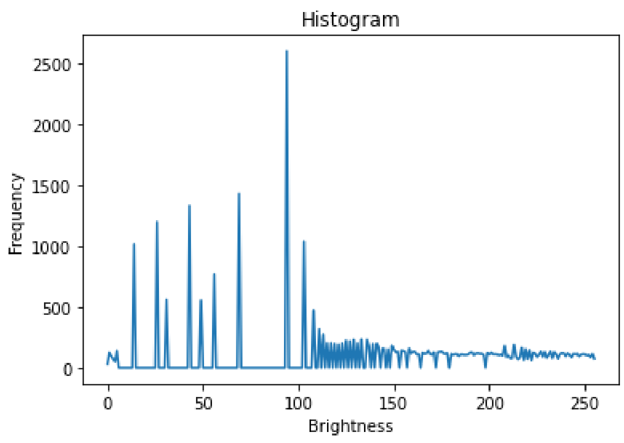


Figure 4.2.4 Histogram after HE

Measures of Entropy, AMBE, PSNR, SSIM for assessing the quality of the image are used. The values of these measures for histogram are as follows

Entropy: 6.3468

AMBE: 65.5673

PSNR: 2.8984

SSIM: 0.7737

4.3.7 GAAHE technique for Input Image2

Image after applying GAAHE(Genetic algorithm based adaptive histogram equalization) technique results in

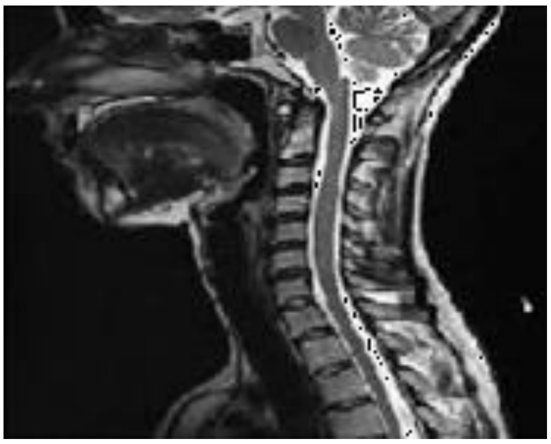


Figure 4.2.5 Equalized Image after GAAHE

Objective Function: -10.48074

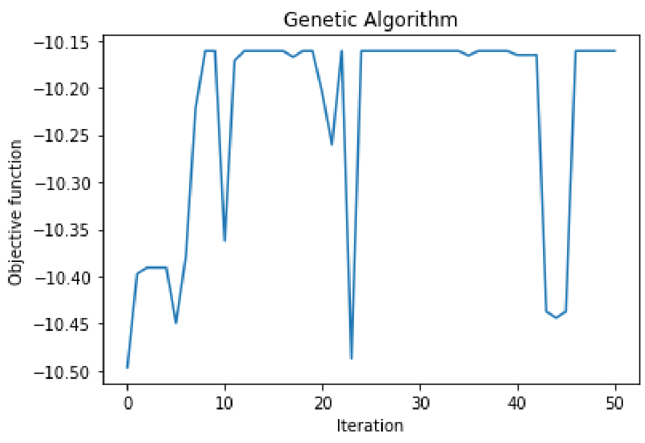


Figure 4.2.6 GA Histogram for Image2 before changing parameters

Histogram for above image is as follows

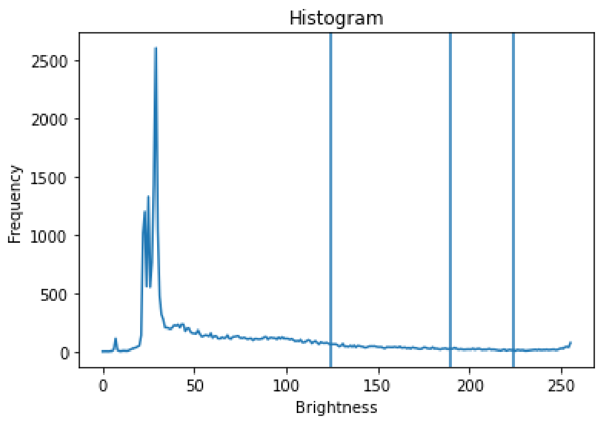


Figure 4.2.7 Histogram after GAAHE

The values of the image assessment measures are as follows for GAAHE,

Entropy: 6.5936

AMBE: 10.1701

PSNR: 23.3145

SSIM: 0.9559

4.3.8 Proposed Method

The input image1 after equalization with the proposed method is as follows

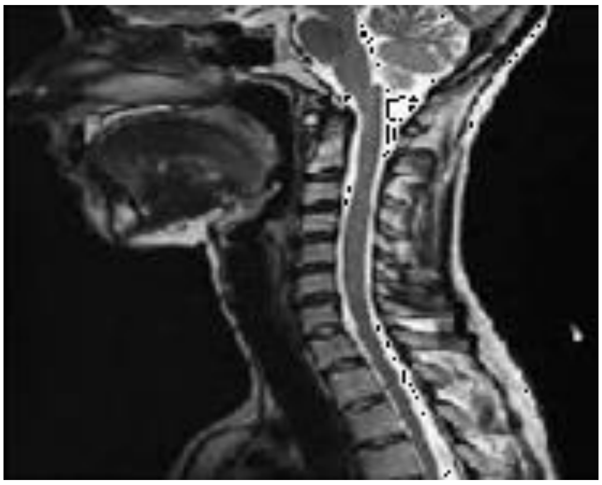


Figure 4.2.8 Equalized Image after applying proposed method

Objective Function: -10.4964

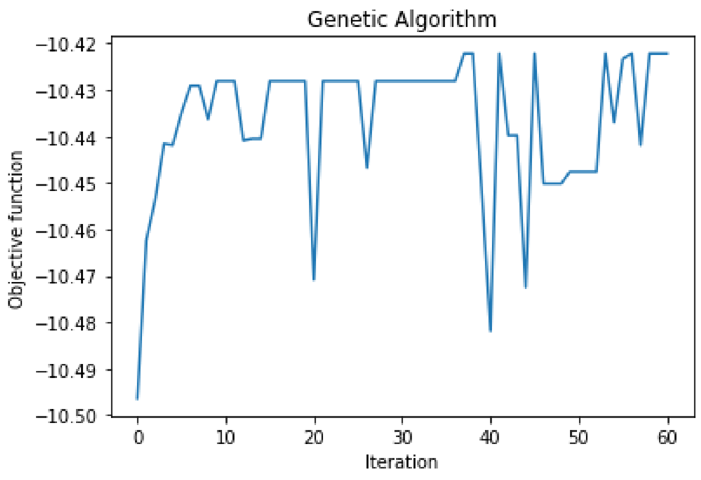


Figure 4.2.9 GA Histogram for Image2 after changing parameters

Histogram for above image is

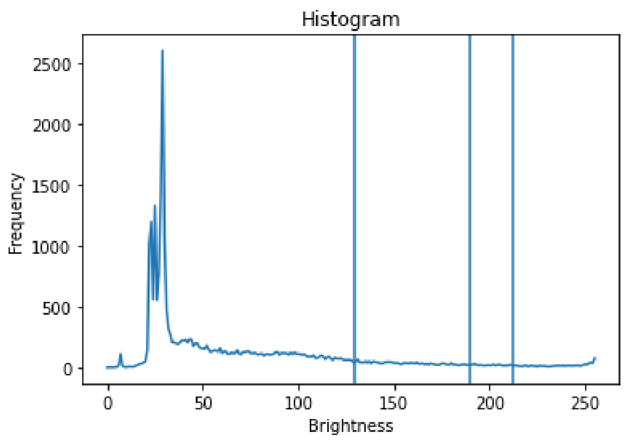


Figure 4.2.10 Histogram after proposed method

The values of the image assessment measures are as follows for proposed method,

Entropy: 5.0326

AMBE: 50.6127

PSNR: 12.9222

SSIM: 0.6632

Enhanced image is obtained using proposed method. To compare proposed method with other method Histogram equalization and GAAHE methods are taken into consideration so that histograms and images can be compared and also function value and entropy can also be compared between these methods.

Parameters are changed in proposed method when compared to GAAHE method. And also in proposed method four objective functions are considered whereas in GAAHE method only three objective functions are considered. There is change in population_size , max_num_iterations, parents_proportion parameters between proposed method and GAAHE method.

5. Conclusion and Future Scope

5.1 Conclusion

By using genetic algorithms to calculate optimal sub exposure thresholds in histogram equalization image is divided into sub images. On these sub images histogram equalization is applied separately to correct the contrast levels of the image. This resulted in enhancement of the medical images which could lead to a better diagnosis. So that, the proposed medical image enhancement technique may be helpful in the interpretation, monitoring and diagnosis of the disease in the human body and proposed technique is more adaptive.

5.2 Future Scope

It is suggested to look into further tweaking the algorithm by tuning the parameters to achieve a better result for certain images. Algorithms other than genetic algorithms maybe found to achieve an even better result. In future this technique may be helpful in the interpretation, monitoring and diagnosis of the disease in the human body.

References

[1] Upendra kumar Acharya, Sandeep Kumar , Genetic algorithm based adaptive histogram equalization (GAAHE) technique for medical image enhancement , Optik 230(2021) 166273.

[2] Upendra kumar Acharya, Sandeep Kumar ,Particle swarm optimized texture based histogram equalization (PSOTHE) for MRI brain image enhancement , Optik 224(2020) 165760.

[3] Pankaj Kandhway, Ashish Kumar, Anurag Singh ,A novel reformed histogram equalization based medical image contrast enhancement using krill herd optimization, Biomed. Signal Process.Control 56(2020) 101677.

[4] Md. Moniruzzaman , Md. Shafuzzaman , Md. Foisal Hossain, Brightness preserving Bi-histogram equalization using edge pixels information, IEEE Trans. on Consum. Electron. 53(2) (2014) 593-600.

[5] S.D. Chen, A.R. Ramli, Minimum mean brightness error bi-histogram equalization in contrast enhancement, IEEE Trans. Consum. Electron. 49 (4) (2003) 1310–1319.

[6] B. Subramani, M. Veluchamy, MRI brain image enhancement using brightness preserving adaptive fuzzy histogram equalization, Int. J. Imaging Syst. Technol. 28 (3) (2018) 217–222.

[7] K. Singh, D.K. Vishwakarma, G.S. Walia, R. Kapoor, Contrast enhancement via texture region based histogram equalization, J. Mod. Opt. 63 (15) (2016) 1444–1450.

[8] T.L. Tan, K.S. Sim, C.P. Tso, Image enhancement using background brightness preserving histogram equalisation, Electron. Lett. 48 (3) (2012) 155–157.

[9] L. Rundo, A. Tangherloni, M.S. Nobile, C. Militello, D. Besozzi, G. Mauri, P. Cazzaniga, MedGA: a novel evolutionary method for image enhancement in medical imaging systems, Expert Syst. Appl. 119 (2019) 387–399.

[10] U.K. Acharya, S. Kumar, Image enhancement using exposure and standard deviation-based sub-image histogram equalization for night-time images, in: Proceedings of International Conference on Artificial Intelligence and Applications, Springer, Singapore, 2020, pp. 607–615.

[11] D. Nirmala, Medical image contrast enhancement techniques, Res. J. Pharm. Biol. Chem. Sci. 6 (3) (2015) 321–329.

[12] V. Anoop, P.R. Bipin, Medical image enhancement by a bilateral filter using optimization technique, J. Med. Syst. 43 (8) (2019) 240.

[13] C.M. Chen, C.C. Chen, M.C. Wu, G. Horng, H.C. Wu, S.H. Hsueh, H.Y. Ho, Automatic contrast enhancement of brain MR images using hierarchical correlation histogram analysis, J. Med. Biol. Eng. 35 (6) (2015) 724–734
[14] Z. Huang, Z. Wang, J. Zhang, Q. Li, Y. Shi, Image enhancement with the preservation of brightness and structures by employing contrast limited dynamic quadrihistogram equalization, Optik (2020), 165877.
[15] Biomedical Image Search Engine https://openi.nlm.nih.gov.

Appendix

Code:

//main.py:

```
import numpy as np
import cv2
from geneticalgorithm import geneticalgorithm as ga
from cdf import calculate_cdfs, calculate_transfer_functions
from clipping import clip_histogram
from equalize import histogram_equalize
from evaluate import evaluate_image
from exposure import calculate_exposure_threshold, max_gray_value
from fitness import calculate_fitness
from histogram import get_histogram
from image_utils import read_gray_scale_image
from pdf import calculate_pdfs_of_sub_images, calculate_modified_pdfs
from plotting_utils import plot_histogram_with_thresholds


def equalize_image(lower_threshold, upper_threshold):
    clipped_histogram = clip_histogram(original_histogram)
    pdfs = calculate_pdfs_of_sub_images(clipped_histogram, lower_threshold,
exposure_threshold,upper_threshold, L)
    modified_pdfs = calculate_modified_pdfs(pdfs)
    cdfs = calculate_cdfs(modified_pdfs)
    transfer_functions = calculate_transfer_functions(cdfs, lower_threshold,
exposure_threshold,upper_threshold, L)
    return histogram_equalize(original_image, transfer_functions, lower_threshold,
                              exposure_threshold, upper_threshold, L)

def fitness_function(exposure_thresholds):
    lt, ut = exposure_thresholds
    resulting_image = equalize_image(lt, ut)
    resulting_histogram = get_histogram(resulting_image)
    return -calculate_fitness(original_image, resulting_image, resulting_histogram
def fitness_gahe_function(exposure_thresholds):
    lt, ut = exposure_thresholds
    resulting_image = equalize_image(lt, ut)
    resulting_histogram = get_histogram(resulting_image)
    return -calculate_fitness(original_image, resulting_image, resulting_histogram)

def equalize_with_fitness(fitness_func, ga_parameters):
    var_bound = np.array([[0, exposure_threshold], [exposure_threshold + 1, L - 1]])
    model = ga(function=fitness_func, dimension=2, variable_type='int',
variable_boundaries=var_bound, algorithm_parameters=ga_parameters)
    model.run()
    lower_exposure_threshold, upper_exposure_threshold =
model.output_dict['variable']
    plot_histogram_with_thresholds(original_histogram,[exposure_threshold,
lower_exposure_threshold, upper_exposure_threshold])
    return lower_exposure_threshold, upper_exposure_threshold,
equalize_image(lower_exposure_threshold,upper_exposure_threshold)

original_image = read_gray_scale_image('images/chest.png')
print(f'Size of original image: {original_image.shape}')
L = int(max_gray_value(original_image))
original_histogram = get_histogram(original_image)
plot_histogram_with_thresholds(original_histogram)
exposure_threshold = int(calculate_exposure_threshold(original_histogram, L))
print(f'Exposure threshold is at : {exposure_threshold}')

## HE
equalized_image_he = cv2.equalizeHist(original_image)
equalized_histogram_he = get_histogram(equalized_image_he)
plot_histogram_with_thresholds(equalized_histogram_he)
cv2.imwrite("equalized_he.png", equalized_image_he)
evaluate_image(original_image, equalized_image_he, original_histogram,
equalized_histogram_he, L)
```

```
## GAAHE
gahe_ga_param = {
    'max_num_iteration': 50,
    'population_size': 50,
    'mutation_probability': 0.01,
    'elit_ratio': 0.0,
    'crossover_probability': 0.8,
    'parents_portion': 0.2,
    'crossover_type': 'uniform',
    'max_iteration_without_improv': None
}


lower_exposure_threshold_gahe, upper_exposure_threshold_gahe,
equalized_image_gahe = equalize_with_fitness(
    fitness_gahe_function, gahe_ga_param)
equalized_histogram_gahe = get_histogram(equalized_image_gahe)
cv2.imwrite("equalized_gaahe.png", equalized_image_gahe)
evaluate_image(original_image, equalized_image_gahe, original_histogram,
equalized_histogram_gahe, L)

## Current
ga_param = {
    'max_num_iteration': 60,
    'population_size': 45,
    'mutation_probability': 0.01,
    'elit_ratio': 0.0,
    'crossover_probability': 0.8,
    'parents_portion': 0.3,
    'crossover_type': 'uniform',
    'max_iteration_without_improv': None
}
lower_exposure_threshold, upper_exposure_threshold, equalized_image =
equalize_with_fitness(fitness_function, ga_param)
```

```
equalized_histogram = get_histogram(equalized_image)
cv2.imwrite("equalizedprop.png", equalized_image)
evaluate_image(original_image, equalized_image, original_histogram,
equalized_histogram, L)
```

//pdf.py:

```
import numpy as np
def calculate_pdf(histogram, lower_bound, upper_bound):
    lower_bound = int(lower_bound)
    upper_bound = int(upper_bound)
    num_elements = (upper_bound - lower_bound) + 1
    num_pixels = 0
    for k in range(lower_bound, upper_bound + 1):
        num_pixels += histogram[k]
    pdf = np.zeros(num_elements, float)
    for k in range(lower_bound, upper_bound + 1):
        if num_pixels == 0:
            pdf[k - lower_bound] = 0.0
        else:
            pdf[k - lower_bound] = float(histogram[k]) / num_pixels
    return pdf

def calculate_pdfs_of_sub_images(histogram, lower_exposure_threshold,
exposure_threshold, upper_exposure_threshold, L):
    first_lower_pdf = calculate_pdf(histogram, 0, lower_exposure_threshold)
    second_lower_pdf = calculate_pdf(histogram, lower_exposure_threshold + 1,
exposure_threshold)
    first_upper_pdf = calculate_pdf(histogram, exposure_threshold + 1,
upper_exposure_threshold)
    second_upper_pdf = calculate_pdf(histogram, upper_exposure_threshold + 1, L - 1)
    return first_lower_pdf, second_lower_pdf, first_upper_pdf, second_upper_pdf

def modify_pdf(p, pdf_sum):
    if pdf_sum + p == 0:
        return 0.0
    else:
        return p / (pdf_sum + p)

def calculate_modified_pdf(pdf):
    pdf_sum = np.sum(pdf)
    return tuple(map(lambda p: modify_pdf(p, pdf_sum), pdf))

def calculate_modified_pdfs(pdfs):
    return tuple(map(calculate_modified_pdf, pdfs))
```

//clipping.py:

```
import numpy as np
def compute_clipping_threshold(histogram):
    c_median = np.median(histogram)
    c_mean = np.mean(histogram)
    return (c_mean + c_median) / 2

def clipper(x, clipping_threshold):
    if x > clipping_threshold:
        return int(clipping_threshold)
    else:
        return x

def clip_histogram(histogram):
    clipping_threshold = compute_clipping_threshold(histogram)
    return np.array([clipper(x, clipping_threshold) for x in histogram])
```

//image_utils.py:

```
import numpy as np
from PIL import Image
from constants import GRAYSCALE_CONVERSION
```

```
def read_gray_scale_image(path):
    return np.array(Image.open(path).convert(GRAYSCALE_CONVERSION))
```

//equalize.py:

```
import numpy as np
def histogram_equalize(image, transfer_functions, lower_exposure_threshold,
exposure_threshold, upper_exposure_threshold, L):
    equalized_image = np.zeros(image.shape, int)
    lower_exposure_threshold = int(lower_exposure_threshold)
    upper_exposure_threshold = int(upper_exposure_threshold)
    for (x, y), b in np.ndenumerate(image):
        if 0 <= b <= lower_exposure_threshold:
            equalized_image[x, y] = int(transfer_functions[0][b])
        elif lower_exposure_threshold + 1 <= b <= exposure_threshold:
            equalized_image[x, y] = int(transfer_functions[1][b -
lower_exposure_threshold - 1])
        elif exposure_threshold + 1 <= b <= upper_exposure_threshold:
            equalized_image[x, y] = int(transfer_functions[2][b - exposure_threshold - 1])
        elif upper_exposure_threshold + 1 <= b <= L - 1:
            equalized_image[x, y] = int(transfer_functions[3][b -
upper_exposure_threshold - 1])
    return equalized_image
```

//fitness.py:

```
import math
import numpy as np
from skimage import filters, feature
from exposure import max_gray_value
from pdf import calculate_pdf

def calculate_fitness(original_image, equalized_image, equalized_histogram):
    cf1 = calculate_texture_parameter_fitness_function(equalized_image,
equalized_histogram)
    cf2 = calculate_edge_parameter_fitness_function(equalized_image)
    cf3 = calculate_psnr(original_image, equalized_image)
    cf4 = calculate_ambe(original_image, equalized_image)
    return (0.25 * cf1) + (0.25 * cf2) + (0.25 * cf3) + (0.25 * cf4)

def calculate_fitness_gahe(original_image, equalized_image, equalized_histogram):
    cf1 = calculate_texture_parameter_fitness_function(equalized_image,
equalized_histogram)
    cf2 = calculate_edge_parameter_fitness_function(equalized_image)
    cf3 = calculate_psnr(original_image, equalized_image)
    return (0.33 * cf1) + (0.33 * cf2) + (0.33 * cf3)

def calculate_ambe(original_image, equalized_image):
    mean_brightness_original = np.mean(original_image)

    mean_brightness_equalized = np.mean(equalized_image)
    return math.fabs(mean_brightness_original - mean_brightness_equalized)

def calculate_mean_squared_error(original_image, equalized_image):
    r, c = original_image.shape
    error = 0
    for (i, j), b in np.ndenumerate(original_image):
        error += (b - equalized_image[i, j]) ** 2
    return (1 / (r * c)) * error

def calculate_psnr(original_image, equalized_image):
    mse = calculate_mean_squared_error(original_image, equalized_image)
    return 10 * math.log((255 ** 2) / mse, 10)

def calculate_entropy(histogram, n):
    pdf = calculate_pdf(histogram, 0, n - 1)
    entropy = 0
    for pd in pdf:
        if pd == 0:
            continue
        entropy += pd * math.log(pd, 2)
    return -entropy

def calculate_texture_parameter_fitness_function(equalized_image,
equalized_histogram):
    glcm = feature.greycomatrix(equalized_image, [1], [0], levels=256)
    contrast = feature.greycoprops(glcm, 'contrast')[0, 0]
    energy = feature.greycoprops(glcm, 'energy')[0, 0]
    n = max_gray_value(equalized_image)
    entropy = calculate_entropy(equalized_histogram, n)
    result = (contrast * math.exp(entropy)) / energy
    if result == 0:
        return 0
    else:
        return math.log(result)

def calculate_edge_parameter_fitness_function(equalized_image):
    sobel = calculate_sobel(equalized_image)
    sum_of_intensity = np.sum(sobel)
    number_of_edges = np.count_nonzero(sobel)
    w, x = equalized_image.shape
    first_term = math.log(sum_of_intensity)
    second_term = (number_of_edges / (w * x))
    if first_term < 0:
        return first_term * second_term
    else:
        return math.log(first_term) * second_term

def calculate_sobel(equalized_image):
    return filters.sobel(equalized_image)
```

//constants.py:

```
MAX_BRIGHTNESS = 255
GRAYSCALE_CONVERSION = 'L'
```

//plotting_utils.py:

```
from matplotlib import pyplot as plt
def add_threshold(ax, threshold):
    ax.axline((threshold, 0), (threshold, 40))
def create_histogram_plot(hist):
    fig = plt.figure()
    ax = fig.add_subplot(111)
    ax.set_title('Histogram')
    ax.set_ylabel('Frequency')
    ax.set_xlabel('Brightness')
    ax.plot(range(len(hist)), hist)
    return ax
def plot_histogram_with_thresholds(hist, thresholds=[]):
    ax = create_histogram_plot(hist)
    for threshold in thresholds:
        add_threshold(ax, threshold)
    plt.show()
```

//cdf.py:

```
import numpy as np
def calculate_cdf(pdf):
    return np.cumsum(pdf)


def calculate_cdfs(pdfs):
    return tuple(map(calculate_cdf, pdfs))


def calculate_transfer_functions(cdfs, lower_exposure_threshold, exposure_threshold,
upper_exposure_threshold, L):
    first_lower_transfer_function = cdfs[0] * lower_exposure_threshold
    second_lower_transfer_function = (lower_exposure_threshold + 1) + cdfs[1] *
(exposure_threshold - (lower_exposure_threshold + 1))
    first_upper_transfer_function = (exposure_threshold + 1) + cdfs[2] *
(upper_exposure_threshold - (exposure_threshold + 1))
```

```
    second_upper_transfer_function = (upper_exposure_threshold + 1) + cdfs[3] * (L -
(upper_exposure_threshold + 1))
    return first_lower_transfer_function,
second_lower_transfer_function,first_upper_transfer_function,
second_upper_transfer_function
```

//histogram.py:

```
import numpy as np
from constants import MAX_BRIGHTNESS
def get_histogram(image):
    histogram = np.zeros(MAX_BRIGHTNESS + 1, int)
    for row in image:
        for value in row:
            histogram[value] += 1
    return histogram
```

//exposure.py:

```
import numpy as np
def max_gray_value(image):
    return np.amax(image)

def calculate_exposure(histogram, L):
    histogram_sum = np.sum(histogram)
    normalize_sum = 0
    for k, h in np.ndenumerate(histogram):
        normalize_sum += k[0] * h
    factor = normalize_sum / histogram_sum
    return (1 / L) * factor

def calculate_exposure_threshold(histogram, L):
    return L * (1 - calculate_exposure(histogram, L))
```

//evaluate.py:

```
from fitness import calculate_entropy, calculate_ambe, calculate_psnr
from image_similarity_measures.quality_metrics import fsim, ssim
def print_comparison(method, original_image_score, equalized_image_score):
    print(f"{method} -> Original image: {original_image_score} Equalized image:
{equalized_image_score}")

def evaluate_image(original_image, equalized_image, original_histogram,
equalized_histogram, L):
    print_comparison("Entropy", calculate_entropy(original_histogram, L),
calculate_entropy(equalized_histogram, L))
    print(f"AMBE between images is {calculate_ambe(original_image,
equalized_image)}")
    print(f"PSNR between images is {calculate_psnr(original_image,
equalized_image)}")
    print(f"SSIM between images is {ssim(original_image, equalized_image)}")
```