An Optimization Approach for Medical Image segmentation using entropy-based Multilevel thresholding

A Project report submitted in partial fulfillment of the requirements for

the award of the degree of

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted by

P. Pavan Kumar(318126512094)	**K. Harshini(318126512081)**
B. Sowjanya(318126512063)	**P.V.S. Teja (318126512093)**

Under the guidance of

Mr.Bibekananda Jena , B.Tech,M.Tech,(Ph.D)
Assistant Professor

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGYAND SCIENCES

(Permanently Affiliated to AU, Approved by AICTE and Accredited by NBA & NAAC with 'A' Grade)

Sangivalasa, Bheemilimandal, Visakhapatnam dist.(A.P)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES

(Permanently Affiliated to AU, Approved by AICTE and Accredited by NBA & NAAC with 'A' Grade)

Sangivalasa, Bheemili mandal, Visakhapatnam dist.(A.P)

PRAGNANAM BRAHMA

ANITS

CERTIFICATE

This is to certify that the project report entitled **"AN OPTIMIZATION APPROACH FOR DIGITAL MEDICAL IMAGE SEGMENTATION"** submitted by **P.Pavan Kumar (318126512094), K.Harshini (318126512081), B.Sowjanya (318126512063), P.V.S.Teja (318126512093)** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Engineering** in **Electronics & Communication Engineering** of Andhra University, Visakhapatnam is a record of bonafide work carried out under my guidance and supervision.

Project Guide

Mr. Bibekananda Jena
M.Tech(Ph.D)
Asst. Professor
Department of E.C.E
ANITS

Assistant Professor
Department of E.C.E.
Anil Neerukonda
Institute of Technology & Sciences
Sangivalasa, Visakhapatnam-531 162

Head of the Department

Dr. V. Rajyalakshmi
M.E,Ph.D.,MIEEE,MIETE,MIE
Professor
Department of E.C.E
ANITS

Head of the Department
Department of E C E
Anil Neerukonda Institute of Technology & Sciences
Sangivalasa-531 162

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our project guides **Mr. Bibekananda Jena,** Asst. professor, Department of Electronics and Communication Engineering, ANITS, for his/her guidance with unsurpassed knowledge and immense encouragement. We are grateful to **Dr. V. Rajyalakshmi**, Head of the Department, Electronics and Communication Engineering, for providing us with the required facilities for the completion of the project work.

We are very much thankful to the **Principal and Management, ANITS, Sangivalasa,** for their encouragement and cooperation to carry out this work.

We express our thanks to all **teaching faculty** of Department of ECE, whose suggestions during reviews helped us in accomplishment of our project. We would like to thank **all non-teaching staff** of the Department of ECE, ANITS for providing great assistance in accomplishment of our project.

We would like to thank our parents, friends, and classmates for their encouragement throughout our project period. At last but not the least, we thank everyone for supporting us directly or indirectly in completing this project successfully.

PROJECT STUDENTS

P.Pavan Kumar (318126512094)

K.Harshini (318126512081)

B.Sowjanya (318126512063)

P.V.S.Teja (318126512093)

CONTENTS

ABSTRACT

Image segmentation plays a key role in image processing that helps in obtaining the representation of images. The traditional bi-level segmentation is simple but analyses only simple images. The main challenge of histogram-based method lies in finding the threshold values that render an optimal segmented representation of the original image. The cross-entropy is a technique proposed to determine the best threshold values by minimizing the cross-entropy between original and segmented images.The classical histogram-based approach for multilevel segmentation is relatively expensive. Although there were many successful approaches still there is room for improvement. It is because of this the evolutionary algorithms like the cluster chaotic optimization (CCO)and Honey Badger Algorithm came into picture. The search strategy that CCO and HBA has taken plays a vital role in the classification procedures of clustering techniques and the randomness of chaotic sequences.

In this project minimum cross-entropy multi-level segmentation using CCO (CEMS-CCO) and HBA (CEMS-HBA) are used to extract the desired region by segmenting different types of medical images lines, brain MRI, Chest X-rays, Blood cell images etc.The segmented results of the above two process are also compared.

CHAPTER-1

INTRODUCTION

1.1OVERVIEW

Image segmentation[1] is a crucial stage in digital image processing used to obtain a more straightforward representation of images. Although classic bi-level segmentation is a relatively simple task, it only suffices to analyse rather simple images. More complex real-life scenarios such as medical imaging processing usually require multi-level segmentation to differentiate between the many regions of interest present in the original images. Traditional histogram-based approaches for multi-level segmentation tend to perform suboptimal, with the best performing being computationally expensive. This difficult compromise between performance and computational cost has led to the proposal of new approaches mixing a variety of optimization algorithms and statistical criteria. Despite the success of these new approaches, there is still room for improvement. It is under these circumstances that evolutionary algorithms like the cluster chaotic optimization (CCO) become relevant. The CCO takes advantage of the classification procedures of clustering techniques and the randomness of chaotic sequences for encouraging the search strategy. This paper proposes a novel method based on the CCO algorithm named minimum cross-entropy multi-level segmentation CCO (CEMS-CCO). The CEMS-CCO employs the cross-entropy as its fitness function and the CCO capabilities to deal with multimodal functions to search for the optimal solution to the multi-level segmentation problem. The CEMS-CCO shows competitive results for medical images multi-level segmentation.

It is more precise in a medical imaging task, wherever the imaging modality[2], imaging conditions, and the organ identity is known. In addition, the pose variations are limited, and there is usually prior knowledge of the number of tissues and the Region of Interest (ROI). On the other side of the coin, the images produced in this field are one of the most challenging due to the poor quality of imaging making the anatomical region segmentation[3] from the background is very incredibly troublesome.

The cluster chaotic optimization (CCO) is a novel approach for solving complex engineering and real-life problems. The CCO is a technique that integrates the classification characteristics of clustering methods and the randomness of chaotic sequences through a synergistic effect for encouraging the search capabilities of solution space. This work presents an accurate multi-level image segmentation approach using the CCO technique to minimize the crossentropy (CEMS-CCO) in order to obtain the optimal threshold values for medical images. To determine the accuracy and performance of the proposed technique, a set of well-known benchmark images is tested within a multilevel segmentation process. In order to corroborate the effectiveness of the proposed approach, different popular optimization techniques are employed to carry out the comparative study. One of them is CEMS-CCO which is already existing and CEMS-HBA is a newly proposed algorithm which outperforms the characteristics of already existing CEMS-CCO algorithm in terms of performance.Specifically, we will consider two types of clustering, one parametric, and the other non-parametric [4]to group pixels into contiguous regions.

The proposed algorithm CEMS-HBA is inspired from the intelligent foraging behaviour of honey badger, to mathematically develop an efficient search strategy for solving optimization problems. The dynamic search behaviour of honey badger with digging and honey finding approaches are formulated into exploration and exploitation phases in HBA. Moreover, with controlled randomization techniques, HBA maintains ample population diversity even towards the end of the search process. To assess the efficiency of HBA, 24 standard benchmark functions, CEC’17 test-suite, and four engineering design problems are solved. There are many optimization algorithms such as Fruit Fly Optimization Algorithm(FFO) [5], Krill Herd Optimization Algorithm(KHO) [6], artificial bee colony (ABC) [7], crow search algorithm (CSA) [8], cuckoo search (CS) [9],etc which are successfully applied for multilevel thresholding[10] in image segmentation application. Recently proposed cluster chaotic optimization algorithms (CCO) [11] found superior over above optimization algorithm on thresholding of images using cross entropy criteria as the fitness function. The experimental results, along with statistical analysis, reveal the effectiveness of HBA for solving optimization problems with complex search-space, as well as, its superiority in terms of convergence speed and exploration–exploitation balance, as compared to other methods used in this study.

1.2 LITERATURE SURVEY

At its core, the search of threshold values for histogrambased segmentation is an optimization problem. For this reason, optimization algorithms (OAs) have become key for the solution of the multi-level thresholding problem. In particular, stochastic OAs, which unlike their deterministic counterpart which tend to get trapped into local minima, deal in a better way with multimodal search spaces. For their part, stochastic techniques take advantage of probabilistic computation techniques to deal with the complexity of multimodal surfaces.

In recent years, several solutions to the multi-level thresholding problem have been incorporated into the literature, bringing together stochastic OAs and different metrics. In 2003, Harikrishna Rai segmented medical images using seeds region growing and gradientbased homogeneity criteria. Akram in 2008employed a gradient-based technique for the recognition of areas of interest by image segmentation. In 2009, Hill used a texture gradient-based watershed transform to perform the image segmentation. Recently, in 2016, Mlakar used a variant of differential evolution to determine the threshold values for image segmentation. In the same year, Rajinikanth mixed between-class variance and the cuckoo search's abilities for image multi-thresholding. In 2017, Pare determined the colour image segmentation employing the hybridization of the firefly algorithm and the Le´vy flights by minimizing the modified fuzzy entropy as the objective function. Dey modified the main structure of some optimization techniques to introduce quantum computing into their operators for image thresholding. In Liang Sheng modified the flower pollination algorithm to overcome its flaws in the exploration and exploitation stages for image multi-thresholding tasks. Agrawal included an edge detection strategy and the coral reef optimization for image segmentation, where the authors report competitive results regarding similar techniques. Shahabi Foroung, in 2019, employed the capabilities of the cuckoo search and the Otsu method for determining the threshold values. Naji modified the slap swarm algorithm operators with the moth-flame optimization, and this adaptation enables a better search strategy, determining better solutions for segmentation applications. Recently, in 2020, Huang changed the strategy of the cuckoo search where each individual shares information with its neighbours to improve the cooperation of the population for enhancing the learning process in the image segmentation. Shahabi employs a recent evolutionary algorithm known as crow search algorithm and the Otsu method for image multi-thresholding and compares the obtained results with some state-of-the-art techniques.

In general, the proposed techniques based on stochastic OAs represent an improvement when compared with traditional approaches both in performance and computational cost. However, considering the complexity of the multi-level thresholding image process and the no free lunch theorem, which indicates that no single OA can determine the optimal solution for all problems, there is still room for improvement. Furthermore, considering that the OAs are developed considering specific characteristics and wellknown testing functions, the test of a different OAs is necessary.

1.3 ORGANIZATION OF THE PROJECT REPORT:

The project report is organized in eight chapters. It starts from the introduction. The current chapter introduces the project and gives a brief description of image segmentation. Chapter 2 provides information about image processing, it’s advantages and applications. Chapter 3 gives a detailed explanation about image segmentation, different approaches in image segmentation and the different techniques that are involved. Chapter 4 gives a detailed explanation about multi-level segmentation using cluster chaotic optimisation. Chapter 5 gives a brief idea about the multi-level segmentation using the Honey Badger Algorithm, it’s general biology, it’s algorithm and the flowchart. Chapter 6 shows us the experimental results and analysis. It contains the results and performance characteristics of the images.Chapter 7 gives the conclusion.Later we have the appendix which contains programs of Multi-level segmentation using CCO and HBA and references of the project.

CHAPTER-2

IMAGE PROCESSING

2.1 INTRODUCTION:

Image processing is a method to perform some operations on an image, in order to get an enhanced image or to extract some useful information from it. It is a type of signal processing in which input is an image and output may be image or characteristics/features associated with that image. Nowadays, image processing is among rapidly growing technologies. It forms core research area within engineering and computer science disciplines too. Fig.1 shows the basic example of Image processing.

Image processing basically includes the following three steps:

1.Importing the image via image acquisition tools;

2.Analysing and manipulating the image;

3.Output in which result can be altered image or report that is based on image analysis.

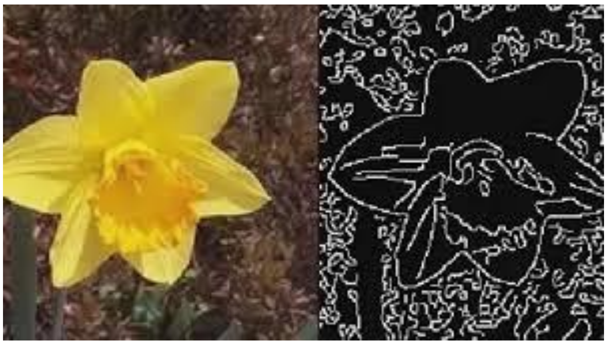


Fig.1 Example of image processing

2.2.1 TYPES OF IMAGE PROCESSING:

There are two types of methods used for image processing namely:

- **Analogue image processing**:

It can be used for the hard copies like printouts and photographs. Image analysts use various fundamentals of interpretation while using these visual techniques.

- **Digital image processing**:

This technique helps in manipulation of the digital images by using computers. The three general phases that all types of data have to undergo while using digital technique are pre-processing, enhancement, and display, information extraction.

2.2 DIGITAL IMAGE PROCESSING TASKS:

Image Reconstruction

• Removal of system or imaging aberrations.

• Aims to reconstruct the best image from collected data.

• Typically output images for visual inspection.

Image Analysis

• Computer analysis of images.

• Extract features or regions

• Recognition of objects.

• High level pattern recognition.

Image Formation

Image formed by computer

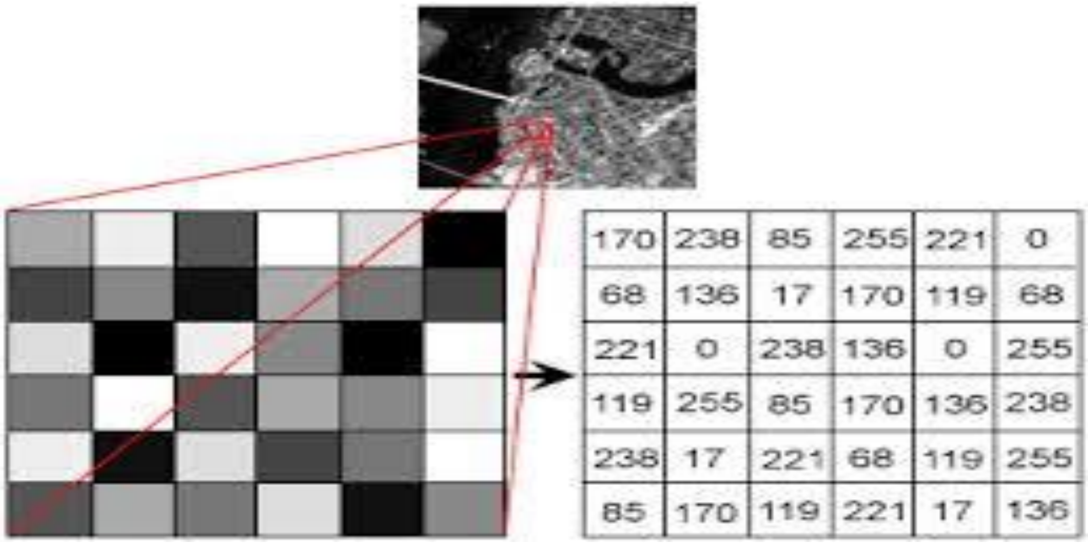
Image Compression and Encoding

• Document and image storage.

Image Representation:

An image defined in the "real world" is considered to be a function of two real variables, for example,f(x,y) with f as the amplitude (e.g. brightness) of the image at the real coordinate position (x,y).

The effect of digitization is shown in Fig.2



170	238	85	255	221	0
68	136	17	170	119	68
221	0	238	136	0	255
119	255	85	170	136	238
238	17	221	68	119	255
85	170	119	221	17	136

Fig. 2Effect of digitization

The 2D continuous image f(x,y) is divided into N rows and M columns. The intersection of a row and a column is called as pixel. The value is assigned to the integer coordinates m,n with { m=0,1,2, ….., M-1}and { n=0,1,2,…..,N-1} is f[m,n]. In fact, in most cases f(x,y) which we might consider to be the physical signal that impinges on the face of a sensor. Typically an image file such as BMP, JPEG, TIFF etc., has some header and picture information. A header usually includes details like format identifier (typically first information), resolution, number of bits/pixel, compression type, etc.

IMAGE PROCESSING:

A. Scaling:

The theme of the technique of magnification is to have a closer view by magnifying or zooming the interested part in the imagery. By reduction, we can bring the unmanageable size of data to a manageable limit. For resampling an image Nearest Neighborhood, Linear, or cubic convolution techniques are used.

B. Magnification:

This is usually done to improve the scale of display for visual interpretation or sometimes to match the scale of one image to another. To magnify an image by a factor of 2, each pixel of the original image is replaced by a block of 2x2 pixels, all with the same brightness value as the original pixel which is described in Fig. 3.

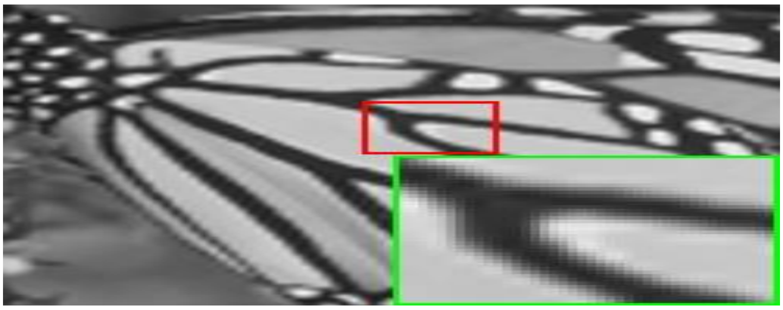


Fig.3Magnification

C. Reduction:

To reduce a digital image to the original data, every$m$thand $n$thcolumn of the original imagery is selected and displayed. Another way of accomplishing the same is by taking the average in ‘m x m’ block and displaying this average after proper rounding of the resultant value.

D. Rotation:

Rotation is used in image mosaic, image registration etc. One of the techniques of rotation in 3-pass shear rotation, where rotation matrix can be decomposed into three separable matrices as shown in below Fig 4.

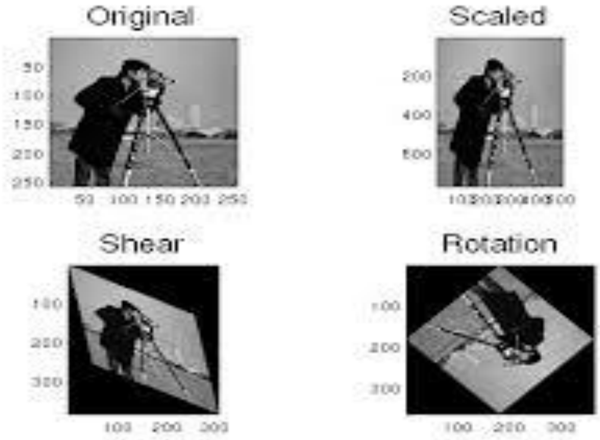


Fig. 4 Scaling, Rotation

E. Mosaic:

Mosaic is a process of combining two or more images to form a single large image without radiometric imbalance. Mosaic is required to get the synoptic view of the entire area, otherwise capture as small images**.**

F. Contrast stretching:

Some Images dense forests, snow, clouds and under lazy conditions over heterogeneous regions)are homogeneous i.e., they do not have much change in their levels. In terms of histogram representation, they are characterized as the occurrence of very narrow peaks. The homogeneity can also be due to the incorrect illumination of the scene.Fig.5 shows us Contrast stretching.

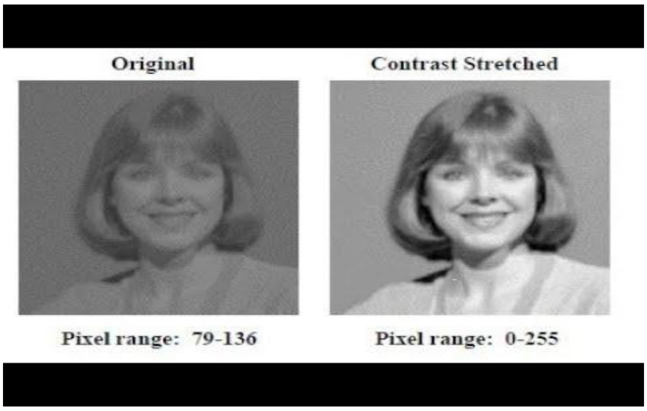


Fig. 5Contrast stretching

I. Noise Filtering:

Noise filtering is used to filter the unnecessary information from an image. It is also used to remove various types of noises from the images. Mostly this feature is interactive. Various filters like low pass, high pass, mean, median etc., are available.

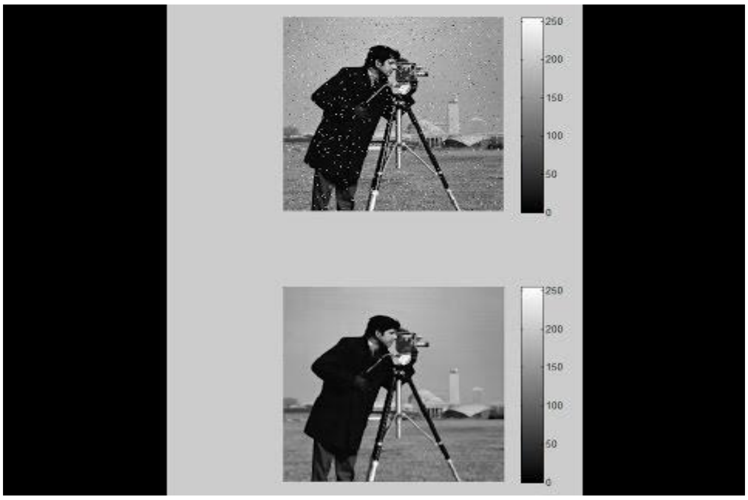


Fig. 6 Noise Filtering

2.3 ADVANTAGES:

Digital image processing in the most layman terms is image editing to improve it's visual appearance but not limited to it. The main advantages of digital image processing are

1. Digital images can be processed by digital computers.
2. Important features such as edges can be extracted from images which can be In industry
3. Images can be given more sharpness and better visual appearance.
4. Minor errors can be rectified.
5. Image sizes can be increased or decreased.
6. Images can be compressed and decompressed for faster image transfer over the network
7. Images can be automatically sorted depending on the contents they have.
8. Unrecognisable features can be made prominent.
9. Images can be smoothened.
10. It allows robots to have vision.
11. It allows industries to remove defective products from the production line.
12. It allows weather forecasting.
13. It is used to analyse cells and their composition.
14. It is used to analyse medical images

2.4 APPLICATIONS:

- Computerized photography (e.g., Photoshop)
- Space image processing (e.g., Hubble space telescope images, interplanetary probe images)
- Medical/Biological image processing (e.g., interpretation of X-ray images, blood/cellular microscope images)
- Automatic character recognition (zip code, license plate recognition)
- Finger print/face/iris recognition
- Remote sensing: aerial and satellite image interpretations
- Reconnaissance
- Industrial applications

CHAPTER-3

IMAGE SEGMENTATION

3.1 INTRODUCTION

Image segmentation is a crucial step in digital image processing, implemented to obtain more straightforward representations of original images. A segmented image is a simplified representation of the original, consisting of several non-overlapping homogeneous regions, facilitating its analysis. For this reason, image segmentation has become instrumental for many computer vision applications, such as satellite imagery, infrared image segmentation, and medical image segmentation. The different segmentation techniques found in the literature can be grouped together into four groups based on their approach to the problem, namely clustering methods, histogram methods, texture analysis methods, and region split and merging methods. The histogram-based approach is the easiest and more widely used approach to the segmentation problem and can be used in two variants: bi-level and multi-level segmentation. During a bi-level segmentation process, the image is segmented into two classes, the foreground and background. Similarly, during multi-level segmentation, the image is segmented into three classes or more. The main challenge of histogram-based methods lies in finding the threshold values that render an optimal segmented representation out of the original image.

Several thresholding techniques have been reported in the literature. One of the most popular techniques is proposed by Otsu, which determines the optimal thresholding value by the maximization of variance between classes. For its part, Tsai, for its part, introduces a moment-preserving principle as an approach for grayscale image segmentation. Another technique extensively used is the Kapur’s method which considers the histogram’s entropy to determine the threshold value. The cross-entropy is a technique proposed by Li to determine the best threshold values by minimizing the cross-entropy between the original and the segmented images. Although all these techniques were originally conceived for a bi-level image segmentation scenario, they easily extend to the multi-level case. However, these changes increase the algorithm’s complexity and, consequently, its computational time.

At its core, the search of threshold values for histogrambased segmentation is an optimization problem. For this reason, optimization algorithms (OAs) have become key for the solution of the multi-level thresholding problem. In particular, stochastic OAs, which unlike their deterministic counterpart which tend to get trapped into local minima, deal in a better way with multimodal search spaces. For their part, stochastic techniques take advantage of probabilistic computation techniques to deal with the complexity of multimodal surfaces. Among the stochastic techniques are those known as evolutionary techniques such as the genetic algorithm (GA), genetic programming (GP) , or swarm intelligence as the particle swarm optimization (PSO), artificial bee colony (ABC), crow search algorithm (CSA), cuckoo search (CS), gravitational search algorithm (GSA), ant colony optimization (ACO), yellow saddle goatfish (YSG), social spider optimization (SSO), mothflame optimization algorithm (MOA).

In recent years, several solutions to the multi-level thresholding problem have been incorporated into the literature, bringing together stochastic OAs and different metrics. In 2003, Harikrishna Rai segmented medical images using seeds region growing and gradientbased homogeneity criteria. Akram in 2008 employed a gradient-based technique for the recognition of areas of interest by image segmentation. In 2009, Hill used a texture gradient-based watershed transform to perform the image segmentation. Recently, in 2016, Mlakar used a variant of differential evolution to determine the threshold values for image segmentation. In the same year, Rajinikanth mixed between-class variance and the cuckoo search's abilities for image multi-thresholding. In 2017, Pare determined the colour image segmentation employing the hybridization of the firefly algorithm and the Le´vy flights by minimizing the modified fuzzy entropy as the objective function. Dey modified the main structure of some optimization techniques to introduce quantum computing into their operators for image thresholding. In, Liang Sheng modified the flower pollination algorithm to overcome its flaws in the exploration and exploitation stages for image multi-thresholding tasks. Agrawal included an edge detection strategy and the coral reef optimization for image segmentation, where the authors report competitive results regarding similar techniques. Shahabi Foroung, in 2019, employed the capabilities of the cuckoo search and the Otsu method for determining the threshold values. Naji modified the slap swarm algorithm operators with the moth-flame optimization, and this adaptation enables a better search strategy, determining better solutions for segmentation applications. Recently, in 2020, Huang changed the strategy of the cuckoo search where each individual shares information with its neighbours to improve the cooperation of the population for enhancing the learning process in the image segmentation. Shahabi employs a recent evolutionary algorithm known as crow search algorithm and the Otsu method for image multi-thresholding and compares the obtained results with some state-of-the-art techniques.

In general, the proposed techniques based on stochastic OAs represent an improvement when compared with traditional approaches both in performance and computational cost. However, considering the complexity of the multi-thresholding image process and the no free lunch theorem, which indicates that no single OA can determine the optimal solution for all problems, there is still room for improvement. Furthermore, considering that the OAs are developed considering specific characteristics and well known testing functions, the test of a different OAs is necessary.

So in the present work we have used two optimisation algorithms,one is CEMS-CCO which is already an existing algorithm in the literature and the other one is newly proposed CEMS-HBA. Then we compared the results obtained for both the algorithms and then we came to a conclusion that CEMS-HBA algorithm is very superior algorithm in all aspects when compared to all the other existing algorithms.

In general, the proposed techniques based on stochastic OAs represent an improvement when compared with traditional approaches both in performance and computational cost. However, considering the complexity of the multi-thresholding image process and the no free lunch theorem, which indicates that no single OA can determine the optimal solution for all problems, there is still room for improvement. Furthermore, considering that the OAs are developed considering specific characteristics and well known testing functions, the test of a different OAs is necessary.

In our work we analysed and got to know that CCO and HBA are the novel OA techniques. Where CCO is used to solve challenging optimization problems with search surfaces that tend to the multimodality[2]. The CCO combines the grouping properties of clustering methods and the stochastic characteristics of chaotic sequences within its search strategy. It is already observed to have a leading performance when determining the optimal values of threshold levels for image segmentation. At each iteration of the CCO’s optimization process, the population is divided into groups (clusters) based on their spatial distribution and then the groups are modified by two operations for adjusting their positions: intra-cluster and extra-cluster. In the Intra-cluster stage, members of the same group are updated by attraction toward the group’s best element; this operation corresponds to the search strategy’s exploitation component. The exploration is covered by the extra-cluster stage, where each group's best-ranked individual is attracted to the overall best agent. In both stages, deterministic operations and chaotic sequences are employed for the position adjustment.At each iteration of the CCO'soptimization process [12], the population is divided into groups (clusters) based on their spatial distribution and then the groups are modified by two operations for adjusting their positions: intra-cluster and extra-cluster. In Fig.7 we can see the basic process of image segmentation.

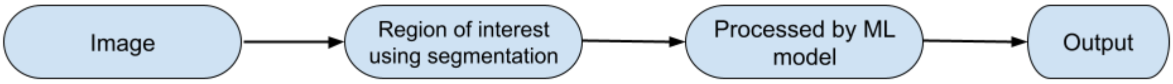


Fig.7Basic process ofSegmentation

3.2APPROACHES IN IMAGE SEGMENTATION [22]

A. **Similarity approach:** This approach is based on detecting similarity between image pixels to form a segment, based on a threshold. ML algorithms like clustering are based on this type of approach to segment an image.

B. **Discontinuity approach:** This approach relies on the discontinuity of pixel intensity values of the image. Line, Point, and Edge Detection techniques use this type of approach for obtaining intermediate segmentation results which can be later processed to obtain the final segmented image.

3.3 IMAGE SEGMENTATION TECHNIQUES

A.Threshold Based Segmentation

B.Edge Based Segmentation

C.Region-Based Segmentation

D.Clustering Based Segmentation

E.Artificial Neural Network Based Segmentation

In this report, we will cover Threshold Based and Edge-based Segmentation. Other segmentation techniques will be discussed in later parts.

A. Threshold Based Segmentation

Image thresholding segmentation is a simple form of image segmentation. It is a way to create a binary or multi-color image based on setting a threshold value on the pixel intensity of the original image.

In this thresholding process, we will consider the intensity histogram of all the pixels in the image. Then we will set a threshold to divide the image into sections. For example, considering image pixels ranging from 0 to 255, we set a threshold of 60. So all the pixels with values less than or equal to 60 will be provided with a value of 0(black) and all the pixels with a value greater than 60 will be provided with a value of 255(white).[22]

Considering an image with a background and an object, we can divide an image into regions based on the intensity of the object and the background. But this threshold has to be perfectly set to segment an image into an object and a background.

Various thresholding techniques are:

i. **Global thresholding**: In this method, we use a bimodal image. A bimodal image is an image with 2 peaks of intensities in the intensity distribution plot. One for the object and one for the background. Then we deduce the threshold value for the entire image and use that global threshold for the whole image. A disadvantage of this type of threshold is that it performs really poorly during poor illumination in the image.

ii. **Adaptive Thresholding**: To overcome the effect of illumination, the image is divided into various subregions, and all these regions are segmented using the threshold value calculated for all these regions. Then these subregions are combined to image the complete segmented image. This helps in reducing the effect of illumination to a certain extent.

iii. **Optimal Thresholding:** Optimal thresholding technique can be used to minimize the misclassification of pixels performed by segmentation.

iv. **Local Adaptive Thresholding:** Due to variation in the illumination of pixels in the image, global thresholding might have difficulty in segmenting the image. Hence the image is divided into smaller subgroups and then adaptive thresholding of those individual groups is done. After individual segmentation of these subgroups, all of them are combined to form the completed segmented image of the original image. Hence, the histogram of subgroups helps in providing better segmentation of the image[22].

B. Edge Based Segmentation

Edge-based segmentation relies on edges found in an image using various edge detection operators. These edges mark image locations of discontinuity in gray levels, color, texture, etc. When we move from one region to another, the gray level may change. So if we can find that discontinuity, we can find that edge. A variety of edge detection operators are available but the resulting image is an intermediate segmentation result and should not be confused with the final segmented image. We have to perform further processing on the image to the segment it. Additional steps include combining edges segments obtained into one segment in order to reduce the number of segments rather than chunks of small borders which might hinder the process of region filling. This is done to obtain a seamless border of the object. The goal of edge segmentation is to get an intermediate segmentation result to which we can apply region-based or any other type of segmentation to get the final segmented image.

TYPES OF EDGES:

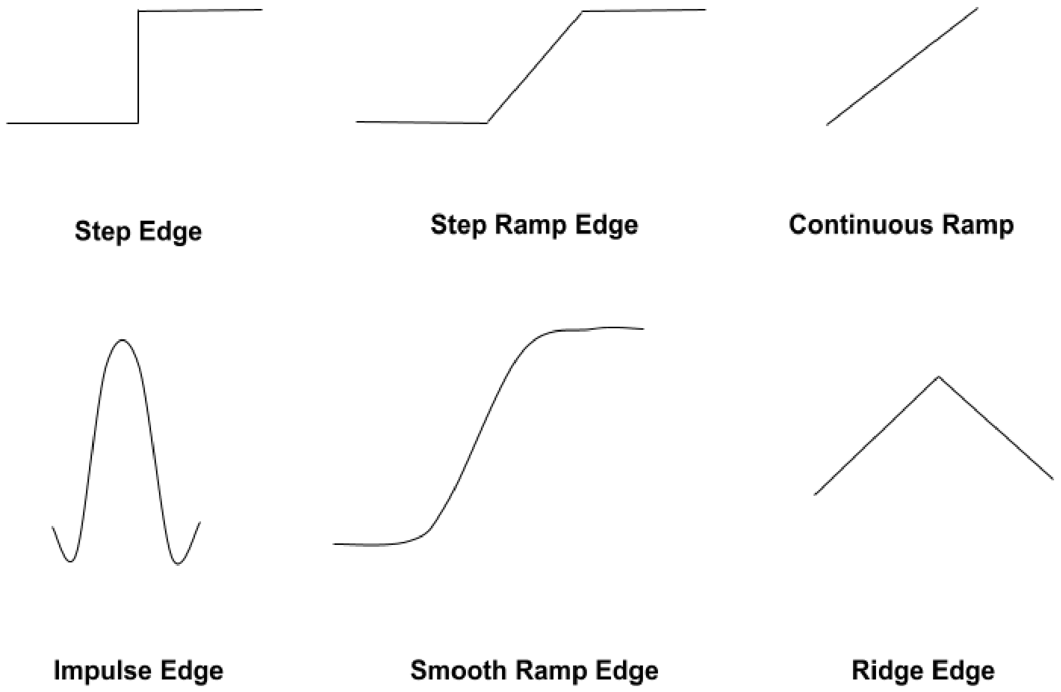


Fig. 8 Types of edges

Edges are usually associated with “Magnitude” and “Direction”. Fig.8 shows us some types of edges. Some edge detectors give both directions and magnitude. We can use various edge detectors like Sobel edge operator, canny edge detector, Kirsch edge operator, Prewitt edge operator, Robert’s edge operator, etc.

C. Region-Based Segmentation[23]

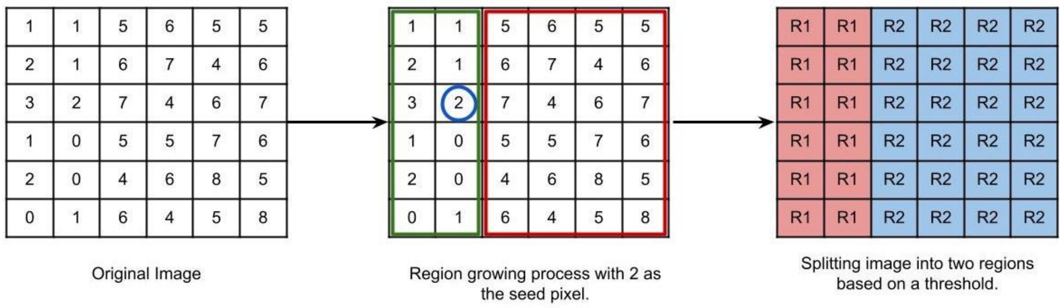
A region can be classified as a group of connected pixels exhibiting similar properties. The similarity between pixels can be in terms of intensity, color, etc. In this type of segmentation, some predefined rules are present which have to be obeyed by a pixel in order to be classified into similar pixel regions. Region-based segmentation methods are preferred over edge-based segmentation methods in case of a noisy image. Region-Based techniques are further classified into 2 types based on the approaches they follow.

i. Region growing method

ii. Region splitting and merging method

i. Region Growing Technique

In the case of the Region growing method, we start with some pixel as the seed pixel and then check the adjacent pixels. If the adjacent pixels abide by the predefined rules, then that pixel is added to the region of the seed pixel and the following process continues till there is no similarity left. This method follows the bottom-up approach. In case of a region growing, the preferred rule can be set as a threshold. For example: Consider a seed pixel of 2 in the given image and a threshold value of 3, if a pixel has a value greater than 3 then it will be considered inside the seed pixel region. Otherwise, it will be considered in another region. Hence 2 regions are formed in the following image based on a threshold value of 3.[23]. The below Fig. 9 shows an example of region growing technique.



1	1	5	6	5	5
2	1	6	7	4	6
3	2	7	4	6	7
1	0	5	5	7	6
2	0	4	6	8	5
0	1	6	4	5	8

Original Image

1	1	5	6	5	5
2	1	6	7	4	6
3	2	7	4	6	7
1	0	5	5	7	6
2	0	4	6	8	5
0	1	6	4	5	8

Region growing process with 2 as the seed pixel.

R1	R1	R2	R2	R2	R2
R1	R1	R2	R2	R2	R2
R1	R1	R2	R2	R2	R2
R1	R1	R2	R2	R2	R2
R1	R1	R2	R2	R2	R2
R1	R1	R2	R2	R2	R2

Splitting image into two regions based on a threshold.

Fig.9 Region growing workflow

ii. Region Splitting and Merging Technique

In Region splitting, the whole image is first taken as a single region. If the region does not follow the predefined rules, then it is further divided into multiple regions (usually 4 quadrants) and then the predefined rules are carried out on those regions in order to decide whether to further subdivide or to classify that as a region. The following process continues till there is no further division of regions required i.e every region follows the predefined rules. In Region merging technique, we consider every pixel as an individual region. We select a region as the seed region to check if adjacent regions are similar based on predefined rules. If they are similar, we merge them into a single region and move ahead in order to build the segmented regions of the whole image. Both region splitting and region merging are iterative processes. Usually, first region splitting is done on an image so as to split an image into maximum regions, and then these regions are merged in order to form a good segmented image of the original image.[23]

In case of Region splitting, the following condition can be checked in order to decide whether to subdivide a region or not. If the absolute value of the difference of the maximum and minimum pixel intensities in a region is less than or equal to a threshold value decided by the user, then the region does not require further splitting. The below Fig.10 shows the workflow of region splitting and merging.

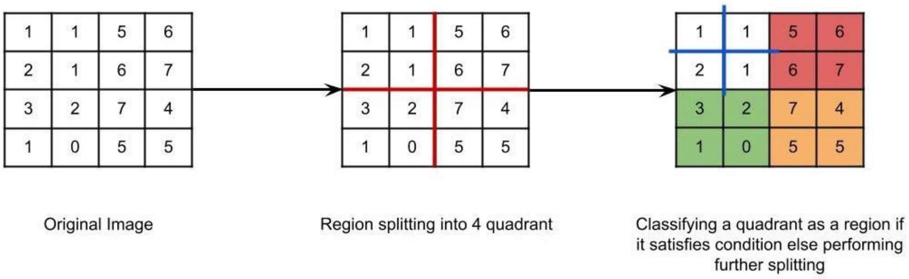

1 1 5 6
2 1 6 7
3 2 7 4
1 0 5 5
Original Image
Region splitting into 4 quadrant
Classifying a quadrant as a region if it satisfies condition else performing further splitting

Fig. 10 Region splitting and merging workflow

(Image by author) Region splitting and merging workflow

$$|Z_{max} - Z_{min}| <= threshold$$

(Image by author)

$$Z_{max} \rightarrow Maximum\ pixel\ intensity\ value\ in\ a\ region.$$

(Image by author)

$$Z_{min} \rightarrow Minimum\ pixel\ intensity\ value\ in\ a\ region.$$

(Image by author)

D. Clustering-Based Segmentation

Clustering is a type of unsupervised machine learning algorithm. It is highly used for the segmentation of images. One of the most dominant clustering-based algorithms used for segmentation is KMeans Clustering. This type of clustering can be used to make segments in a coloured image.

i. KMeans Clustering:

Let's imagine a 2-dimensional dataset for better visualization. First, in the dataset, centroids (chosen by the user) are first randomly initialized. Then the distance of all the points to all the clusters is calculated and the point is assigned to the cluster with the least distance. Then centroids of all the clusters are recalculated by taking the mean of that cluster as the centroid. Then again data points are assigned to those clusters. And the process continues till the algorithm converges to a good solution. Usually, the algorithm takes a very small number of iterations to converge to a solution and does not bounce. Fig.11 and Fig.12 shows the workflow of KMeans clustering.

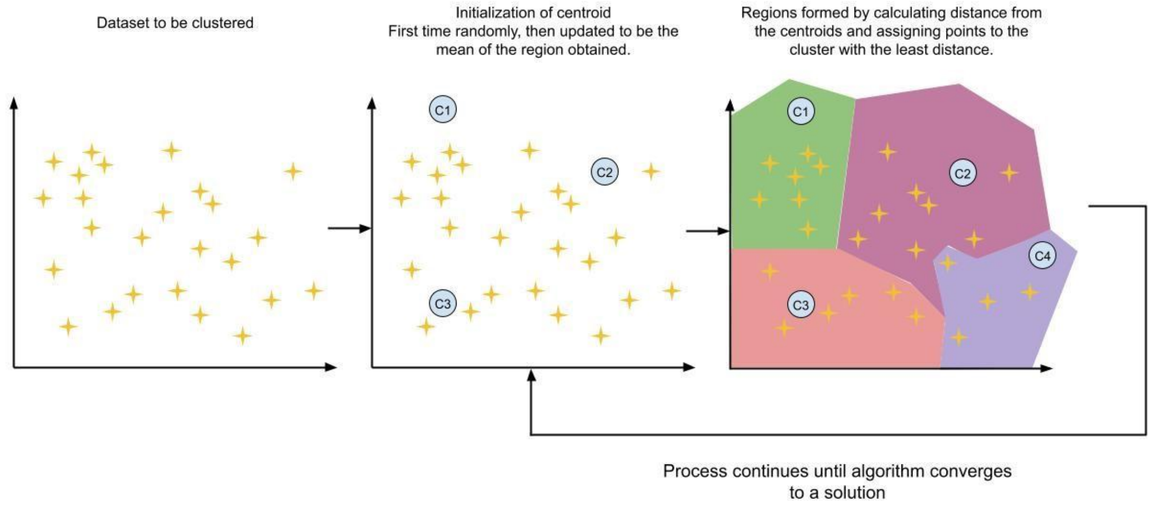

Dataset to be clustered
Initialization of centroid
First time randomly, then updated to be the mean of the region obtained.
Regions formed by calculating distance from the centroids and assigning points to the cluster with the least distance.
C1
C2
C3
C4
Process continues until algorithm converges to a solution

Fig. 11 KMeans clustering workflow

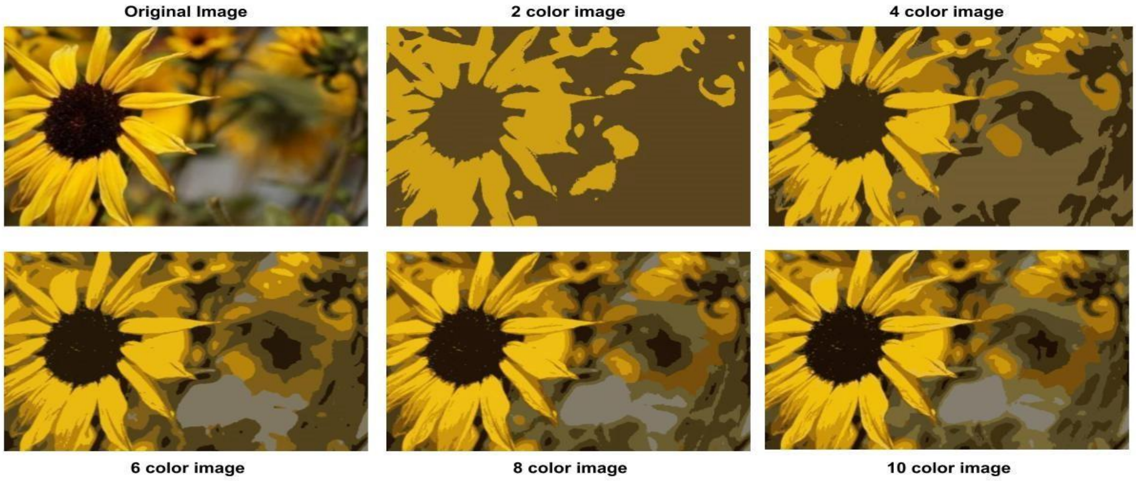


Fig.12 Clustering is performed on an image with a color resembling the number of clusters inputted in the KMeans algorithm.

3.4 ENTROPY BASED MULTILEVEL SEGMENTATION:

Image segmentation is a crucial step in image processing to obtain a simplified representation of the original image easing its analysis. Although many segmentation techniques are reported in the literature, many are quite complex and require high computational resources.

Thresholding is one of the most used methods for image segmentation. It consists of finding an intensity value to separate the grayscale image histogram into two classes, background and foreground. Multi-level thresholding on the other hand, deals with images containing multiple objects with similar grayscale levels. Thus, the correct fitting of thresholding values becomes a more significant challenge. Various techniques have been proposed for determining high accurate threshold values to the multilevel segmentation problem by considering the quality of some criteria. However, most of them only determine suboptimal solutions at a high computational cost. One of the most common quality criteria and the employed in this work is the cross-entropy, which can be described as follows.

3.4.1 CROSS ENTROPY CRITERIA:

The cross-entropy criteria (CE)[12] were first introduced by Li and Lee .It attempts to reduce cross-entropy between the original picture $\mathbf{I}$ and the segmented image $\mathbf{I}_{th}$, A low cross-entropy number indicates greater homogeneity or lower uncertainty .This criterion employs the image's histogram $h(i)$, $i = 1,...,N$, where N is the number of gray levels, and the threshold value utilized to determine the segmented image $\mathbf{I}_{th}$ is as follows:

$$I_{th}(x,y) = \begin{cases} \mu(1,th), & I(x,y) < th, \\ \mu(th, L+1), & I(x,y) \geq Th, \end{cases} (1)$$

where

$$\mu(a,b) = \sum_{i=a}^{b-1} ih(i) / \sum_{i=a}^{b-1} h(i), \qquad (2)$$

Using the following formulation, the objective function (cost function) is computed from the segmented image to obtain the cross-entropy value:

$$f_{ce}(th) = \sum_{i=1}^{th-1} ih(i) \log\left(\frac{i}{\mu(1,th)}\right) + \sum_{i=th}^{L} ih(i) \log\left(\frac{i}{\mu(th,L+1)}\right), (3)$$

Although Eq. (3) depicts bi-level segmentation, it may be expanded to multi-level segmentation at the cost of a significant increase in processing time. Yin[13] devised a strategy to address this problem. Which provides a recursive programming technique for image segmentation that reduces computational time, allowing the statement (3) to be recast as follows:

$$f_{ce}(th) = \sum_{i=1}^{L} ih(i) \log(i) - \sum_{i=1}^{th-1} ih(i) \log\big(\mu(1,th)\big) - \sum_{i=th}^{L} ih(i) \log\big(\mu(th, L+1)\big), (4)$$

the threshold value is replaced by a vector $\mathbf{th} = [th_1, th_2, \dots, th_N]$that includes the N possible combinations of threshold values in multi-level segmentation, and the objective function can be redefined as follows:

$$f_{ce}(\boldsymbol{TH}) = \sum_{i=1}^{L} ih(i) \log(i) - \sum_{i=1}^{m} H_i, \qquad (5)$$

where m denotes the total number of thresholds, and H_i denotes the height of the threshold:

$$H_1 = \sum_{i=1}^{th_1-1} ih(i) \log(\mu(1, th_1)),$$

$$H_k = \sum_{i=th_{k-1}}^{th_1-1} ih(i) \log\big(\mu(th_{k-1}, th_k)\big), \qquad 1 < k < m, \quad (6)$$

$$H_m = \sum_{i=th_m}^{L} ih(i) \log\big(\mu(th_m, L+1)\big),$$

CHAPTER 4

MULTI-LEVEL SEGMENTATION USING CLUSTER CHAOTIC OPTIMIZATION (CCO):

4.1 INTRODUCTION

CCO (cluster chaotic optimization algorithm)[11] is a brand-new optimization technique introduced by Galvez et al. It implements a search strategy for difficult optimization problems by combining the properties of clustering classification techniques and the behaviour of chaotic sequences. The CCO's fundamental idea is to use the capabilities of clustering methods for data processing, as well as the Ward approach, to create a procedure in which the particles pay special attention to locations with relevant information. Here CCO initializes a 'k' number of population of$C^k = \{(c_1^k, c_2^k, \ldots, c_M^k)\}$($k$ runs from 0 to max-iteration), where each population consists of 'M' no of threshold values. Each particle $c_i^k, (i \in 1,2, \ldots, N)$ is an n-dimensional vector $\{c_{i,1}^k, c_{i,2}^k, \ldots, c_{i,m}^k\}$, with each dimension representing one of the problem's decision variables. The CCO begins by generating the population with M uniformly distributed random points in the search space between the upper and lower boundaries, as shown below.

$$c_{i,j}^{k} = l_j + rand(0,1).(u_j - l_j)$$

Where j=1,2,3,...,n; i=1,2,….,N

After the initialization, the population is divided into clusters $g_p^k(p \in 1, \dots, q)$, each of which comprises p elements. All groups are conducted by two primary behaviors, intra-cluster and extra-cluster operators[12], after they have been classified using such clustering procedures.

4.2INTRA-CLUSTER STAGE:

Here all groups are treated independently and processed through the operators like local attraction and local perturbation. The best element of the cluster in terms of fitness quality is regarded in the local attraction, and the attraction generated in all members of the group modifies their locations, it is given by:

$$c_{i,j}^{k+1} = c_{i,j}^{k} + p_{g_p}^{k}.z.\left(c_{b,j}^{k} - c_{i,j}^{k}\right), \quad (8)$$

Where $p_{g_p}^{k} = \frac{|g_p^k|}{N}$ (Eq.(9)) is the cluster density and z (z ∈ [-1, 1]) obtained by using chaotic map.

The perturbation is thought to improve the cluster's search capability.

$h_{i,j}^{A} = c_{i,j}^{k+1} + \left(c_{i,j}^{k+1}.z_A.v_A\right)$,(10)

$$h_{i,j}^{B} = c_{i,j}^{k+1} - \left(c_{i,j}^{k+1}.z_B.v_B\right),$$

where z_A and z_B are the chaotic map's numbers, and v_A and v_B are the circular neighbourhoods whose computation is: $v_l = \cos(\alpha\,.\,r\,)$; *l*= *A*, *B*, where *r* is a random value between$[0,2\pi]$ and α is a perturbation that changes as the loop progresses. Following the inter-cluster change, an elitist criterion determines which members of the cluster are retained and which are deleted in the minimization situation.

$$\boldsymbol{c}_i^{k+1} = \begin{cases} \mathbf{h}_i^A \; if \; J\left(\boldsymbol{h}_i^A\right) < J\left(\boldsymbol{c}_i^{k+1}\right), \\ \boldsymbol{h}_i^B \; if \; J\left(\boldsymbol{h}_i^B\right) < J\left(\boldsymbol{c}_i^{k+1}\right), \; (11) \\ \boldsymbol{c}_i^{k+1} \; otherwise, \end{cases}$$

4.3 EXTRA-CLUSTER STAGE:

At this point, only the best members of the current iteration are considered for movement operations. The same two procedures are used as in the intra-cluster stage: global attraction and global perturbation. Individuals adjust their positions in the global attraction using the following equation:

$\mathbf{c}_{b,j}^{k+1} = \mathbf{c}_{b,j}^{k} + \left(\mathbf{c}_{B,j}^{k+1} + \mathbf{c}_{b,j}^{k}\right).rand.v_G$,(12)

where rand is a random value between 0 and 1, and v_G is the circular neighborhood given in Eq (10). The global perturbation is then calculated using the following formula:

$$h_{b,j}^{r} = c_{b,j}^{k+1} + \left(c_{b,j}^{k+1}.rand.v_r\right), h_{b,j}^{s}$$

$= c_{b,j}^{k+1} - \left(c_{b,j}^{k+1}.rand.v_s\right)$,(13)

where rand and v_r and v_s are the circular neighborhoods specified in Eq (10). Finally, an elitism basis such as Eq. (11) is employed to decide which elements predominate between$\mathbf{h}_i^r$, $\mathbf{h}_i^s$, and $\mathbf{c}_i^{k+1}$.

The below Algorithm summarizes the entire computing operation.

4.4 CEMS-CCO ALGORITHM :

1. Clear memory
2. Input data: ***N, max_gen***, and***n***
3. Initialization: $\mathbf{C}_{\boldsymbol{i}}^{\boldsymbol{k}}(\boldsymbol{i} = \mathbf{1}, \dots, \boldsymbol{N})$
4. While (stop criteria) do
5. Evaluate the quality of the individuals $\boldsymbol{f}(\mathbf{C}_{\boldsymbol{i}}^{\boldsymbol{k}})$
6. Set the best element $(\mathbf{c}_{\boldsymbol{b}}^{\boldsymbol{k}})$
7. Cluster $\mathbf{C}_{\boldsymbol{i}}^{\boldsymbol{k}}(\boldsymbol{g}_{\boldsymbol{p}}^{\boldsymbol{k}}, \boldsymbol{p})$
8. Compute the perturbation$\boldsymbol{\alpha}$
9. for (each group)
10. Determine the best element of each group
11. Calculate local attraction (8)
12. Calculate global perturbation (10)

```
    end for
    for (all best group elements)
        Calculate global attraction (12)
        Calculate global perturbation (13)
    end for
end while
Display $\mathbf{c}_{\boldsymbol{B}}^{\boldsymbol{k}}$
```

CEMS-CCO:

The CEMS-CCO enhanced multi-level digital image segmentation is provided in this paper. The proposed CEMS-CCO uses the attributes of clustering classification methods and concepts of cross entropy multilevel segmentation to formulate a method which will solve optimization problems. Data analysis concepts of machine learning is used. The CCO is used to find the best threshold values by utilizing a search strategy based on data analysis within the search space using machine learning tools such as clustering concepts. To apply the concepts of CCO and obtaining the optimum solution we need a input grayscale image and also the histogram of input image to segment the image using thresholds. So to achieve this a grayscale image **I**is taken as input and then its histogram is calculated using matlab. Then we must give size of the population, the number of iterations, and the dimension of the problem as input parameters to the algorithm to generate initial population. To evaluate the quality of population, a cost function f($\mathbf{C}_i^k$) is used which will evaluate the performance of the algorithm for the given input images. We have to either minimize or maximize the cost function to tell whether obtained solution is a optimized one or not. The population is divided into groups or clusters($\mathbf{C}_i^k$), and then intra-cluster and extra-cluster operations are performed. The algorithm is made to run until a stop criterion is reached. When the stop criterion is reached, then the optimized solution out of all the solutions that we obtain till now is displayed and this is the best solution. Then the final segmented image is obtained with the best optimum values that we have obtained from the algorithm. The complete steps for the multilevel segmentation is shown in the figure.

Step 1. Read a gray scale image **I**.

Step 2. Determine its histogram $h_{\mathbf{I}}$.

Step 3. Initialize the population $\mathbf{C}_i^k$ using the setup parameters, *N*, *max_gen*, and *n* (7).

Step 4. Evaluate the population using the cost function$f(\mathbf{C}_i^k)$ (5).

Step 5. Determine the best individual $\mathbf{c}_b^k$.

Step 6. Cluster the population $\mathbf{C}_i^k(g_p^k, p)$.

Step 7. Compute the intra-cluster and extra-cluster operations (8), (10), (12), and (13).

Step 8. If the *max_gen* is reached, display the best solution $\mathbf{c}_B^k$, otherwise, return from step 5.

Step 9. Generate the resulting image $\mathbf{I}_{seg}$ by using the Equation (15).

4.5CEMS-CCO PROCEDURE:

In the CEMS-CCO procedure, the number of thresholds employed in the segmentation process is represented as the decision variables for each individual in the population and is expressed as $\mathbf{C}_i^k = [\mathbf{th}_1, \mathbf{th}_2, \dots, \mathbf{th}_N]$, $\mathbf{th}_n = [th_1, th_2, \dots, th_n]^T$, where *N* is the population number, *k* is the current iteration, and *n* is the number of thresholds or levels. The CEMS-CCO uses as cost function the cross-entropy. The following flowchart is described in Fig. 13

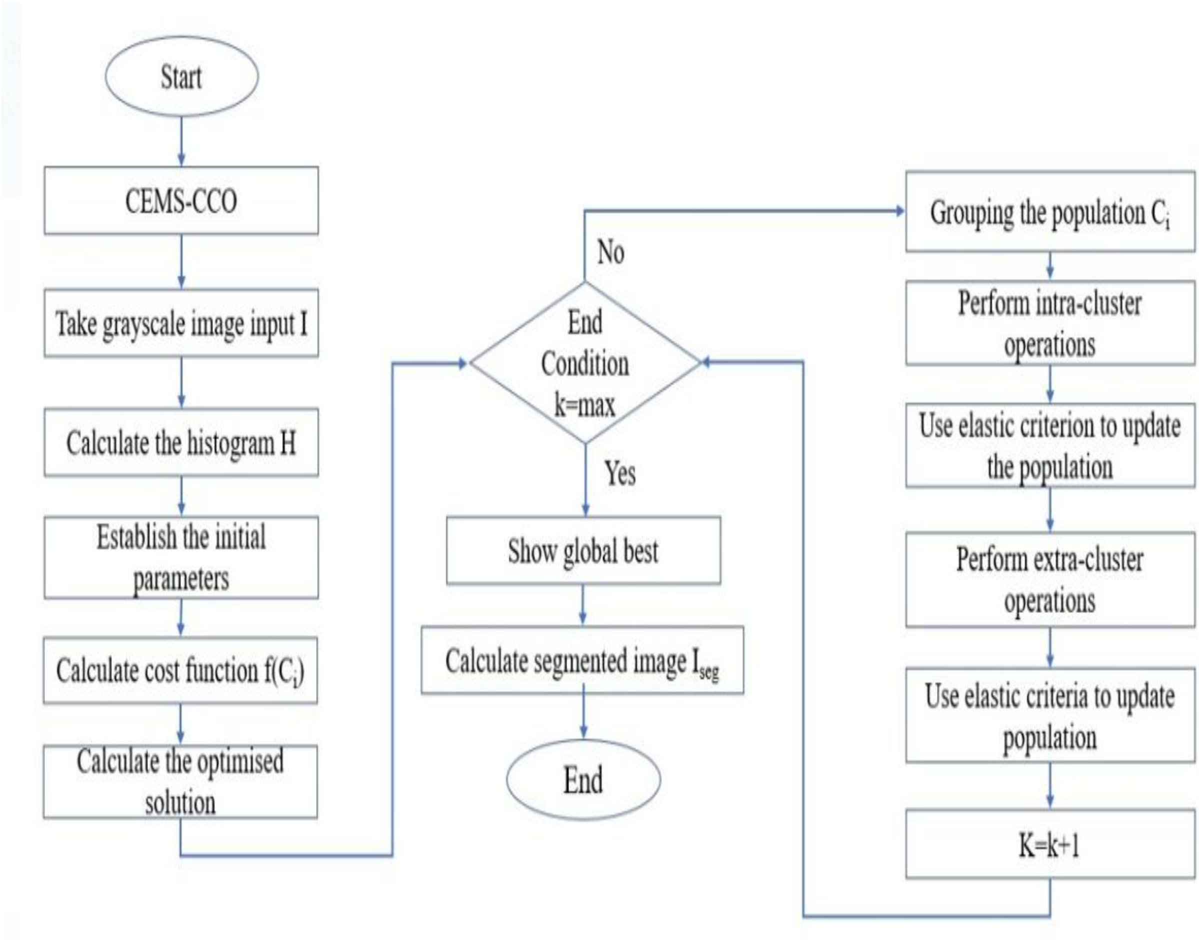

Start
CEMS-CCO
Take grayscale image input I
Calculate the histogram H
Establish the initial parameters
Calculate cost function f(C$_i$)
Calculate the optimised solution
End Condition k=max
No
Yes
Show global best
Calculate segmented image I$_{seg}$
End
Grouping the population C$_i$
Perform intra-cluster operations
Use elastic criterion to update the population
Perform extra-cluster operations
Use elastic criteria to update population
K=k+1

Fig.13 Flowchart of CEMS-CCO approch

Since the multi-level segmentation can be implemented as an optimization approach, the cost function used for such a purpose is the cross-entropy criterion which is defined as:

$$\arg\ \min f_{ce}(\mathbf{th}) \qquad (14)$$

subject to $\mathbf{th} \in \mathbf{C}$

where $f_{ce}(\mathbf{th})$ is the value of the cross-entropy criterion and $\mathbf{C} = \{th\ \in\ \mathrm{R}^n | 0 \le 255, i = 1, \dots, n\}$. After the optimization process, the CCO determines the best solution $\mathbf{c}_B^k$ by minimizing the cost function which is used to compute the segmented image by the following rules:

$$\mathbf{I}_{seg}(r,s) = \begin{cases} \mathbf{I}(\mathbf{r},\mathbf{s}), \text{if} & \mathbf{I}(r,s) \le th_1 \\ th_{i-1}, \text{if} th_{i-1} < \mathbf{I}(r,s) \le th_i, i = 2, \dots, n-1 (15) \\ \mathbf{I}(\mathbf{r},\mathbf{s}), \text{if} & \mathbf{I}(r,s) \le th_n \end{cases}$$

where $\mathbf{I}(r,s)$ is the grayscale value of the segmented image at position r, s, and th_i is the i-threshold value.

CHAPTER -5

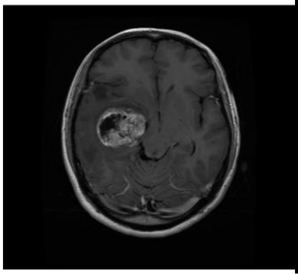
MULTI-LEVEL SEGMENTATION USING HONEY BADGER ALGORITHM:

5.1 INTRODUCTION:[14]

Optimization refers to the process of finding best solutions for a given system from all the possible values to maximize or minimize the output. Over the last few decades, as the complexity of problems has increased, the need for new optimization techniques has become imperious. Earlier, the conventional mathematical techniques that have been used for solving optimization problems are mostly deterministic that suffer from one major problem: local optima entrapment. This makes these techniques highly inefficient in solving real optimization problems, leading to a growing interest in stochastic optimization techniques over the last two decades. Often, most of the real-world optimization problems, in the area of engineering, wireless sensor networks, image processing, feature selection, tuning of machine learning parameters, bio-informatics etc., are highly non-linear and non-convex due to inherent complex constraints and many design variables. Therefore, solving these types of optimization problems is complex because of many inherent local minima. Moreover, there is no guarantee of finding a global solution .Thus, the difficulties associated with these types of real-life optimization problems motivate to develop alternative and effective techniques for better solutions.

In order to find better solution, many researchers have tried to propose new algorithms and/or improved the existing methods. Metaheuristic research community has implemented useful search strategies to obtain the global optimum. Because, in real-life optimization problems, search-space grows exponentially and makes the problem landscape highly multimodal, the conventional optimization methods often produce suboptimal solutions. This has, over the past few decades, led to the development of many new metaheuristic algorithms. These methods have shown robust performances on a wider range of complex problems for obtaining the optimal solutions.

Recently, various search strategies have been effectively incorporated in metaheuristic algorithms; mostly inspired from nature, simulating principles of biology, physics, ethology or swarm intelligence. Interestingly, some of them such as Genetic Algorithm, particle swarm optimization (PSO), and Archimedes optimization algorithm are fairly well-known among not only computer scientists but also scholars from other domains. This has resulted in extensive theoretical work and practical applications using metaheuristic techniques — mainly because of several major reasons including flexibility, gradient-free mechanism, and local optima avoidance. As these methods are gradient-free, there is no need to calculate derivative of the search-space; hence reducing computational cost, being highly flexible for solving a diverse range of problems. Due to these advantages, the application of metaheuristics can be found in different branches of science and industry.[14]



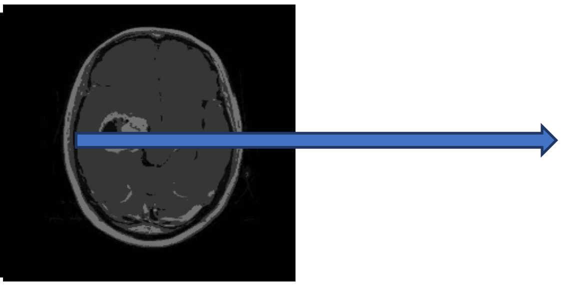


Fig. 14 **Fig. 15**

Input image(beforesegmentation)Output image(after segmentation)

Fig. 14is the input image while Fig. 15 is the segmented image to which we have applied the optimization algorithm. The metaheuristics algorithms are categorized into two main classes: single solution-based and population-based approaches. The literature has evidenced population-based algorithms having better ability to explore the search space, and exploit the global optimum, as compared to single solution-based algorithms. The populationbased algorithms, according to the sources of inspiration, can be divided into three main categories: (1) Swarm Intelligence algorithms (SI), includes swarm-based techniques that mimic the social behaviour of insect or animals groups. (2) Evolutionary Algorithms (EAs), which follow natural evolution process found in nature. And, (3) Natural Phenomenon algorithms (NP) that imitate the physical and chemistry principles; while, some include those inspired by human behaviour, but are neither SI nor EA. The SI metaheuristic algorithms mimic the self-organized and collective behaviours in nature. These algorithms take inspirations from the social behaviour of animals, birds, plants, and human. Some of the well-known metaheuristic algorithms are: Grasshopper Optimization Algorithm (GOA)[24], Whale Optimization Algorithm (WOA)[25], Elephant Herding Optimization (EHO)[26].On the other hand, EAs are a type of stochastic global optimization methods inspired by natural evolution and genetic mechanisms , such as Genetic Algorithm (GAs), Evolution Strategy (ES) , Covariance Matrix

Adaptation Evolutionary Strategy (CMA-ES), and History-based Adaptive Differential Evolution variants with linear population size reduction (L-SHADE). NP algorithms imitate the physical or chemical rules in the universe. Some of the popular and recent algorithms are Simulated Annealing (SA), Thermal Exchange Optimization (TEO) , and Henry Gas Solubility Optimization.[14]

Despite the need for more function evaluations, the literature shows that population-based algorithms are highly suitable for solving real challenging problems. Logically, No-Free-Lunch (NFL) theorem states that there are either no metaheuristic optimization algorithm able to solve all optimization problems or still problems not yet solved. These two reasons are the massive motivation to present a novel metaheuristic algorithm called Honey Badger Algorithm (HBA) which mimics foraging behavior of honey badger[15]. Because the ability to maintain the trade-off balance between exploration and exploitation plays a significant role in effective search, HBA encapsulates dynamic search strategies. This characteristic enables HBA in solving hard optimization problems with many local regions, as it keeps ample population diversity throughout search process for investigating a large area the given landscape. Furthermore, 24 standard mathematical optimization problems, CEC’17 test-suite and, four real-world engineering design optimization problems are solved. The comparison with ten established metaheuristic algorithms, including SA, PSO, CMA-ES, L-SHADE, MFO, EHO, WOA, GOA, TEO, and HHO, validates efficacy of the proposed HBA algorithm.

Fig.16HONEY BADGER

Finally, the following are the papers major contributions:

1. We propose HBA, a new swarm-based optimization technique that replicates Honey Badger's behaviour.
2. HBA is compared to state-of-the-art metaheuristic algorithms in terms of statistical significance, convergence time, exploitation–exploration ratio, and diversity.
3. We conduct a series of experiments to examine the influence of the proposed algorithm's performance on benchmark optimization problems, the CEC'17 test-suite, and real-world engineering design challenges, which are regarded as difficult test-suites in the related literature.

4. On difficult optimization situations, HBA beats other competitor algorithms.

The remainder of the paper is structured as follows: The HBA algorithm's inspiration and mathematical model are presented in Section 2. Section 3 describes the experiments on standard benchmark and CEC'17 tasks, as well as the relative findings. The experimental execution on engineering design challenges is highlighted in Section 4, along with the presentation of the relevant outcomes. Section 5 goes into greater detail about HBA's performance in comparison to a number of other metaheuristic algorithms. Finally, Section 6 brings the paper to a close and offers future research topics.

5.2 HONEY BADGER GENERAL BIOLOGY

The idea behind Honey Badger Algorithm (HBA), which resembles honey badger behaviour in nature, is discussed in this section.

Honey badgers are a bold mammal with black and white fluffy fur that can be found in semi-deserts and rainforests throughout Africa, Southwest Asia, and the Indian subcontinent. This dog's size ranges from 60 to 77 centimeters.A daring forager with a body length of 7 to 13 kilograms and a weight of 7 to 13 kilograms, preys on sixty different species, including the deadlysnakes. It is a clever mammal that can utilize tools and enjoys honey. It prefers to live alone in self-made holes and only meets up with the other badgers to mate. Honey badgers are divided into 12 subspecies. There isn't anything specific.[15]

Honey badgers have a year-round breeding season since cubs are born all year. It is because of their bravery that When it can't get away, it never hesitates to fight even much larger predators This animal can also easily climb trees, to reach food sources such as bird nests and beehives.

A honey badger uses its smelling mice skills to seek its meal by travelling slowly and persistently. By digging and eventually catching the prey, you can get a good idea of where it is. It may dig up to fifty holes in a single day.Foraging attempts have resulted in holes in a radius of forty kilometres or more. Honey badgers enjoy it, but it is not beneficial for them.

When it comes to locating beehives Honey-guide (a bird), on the other hand, can find the hives but not the honey. These events lead to a partnership between the two, in which the bird directs the badger to beehives and assists it in opening them with its long claws, and both reap the benefits of teamwork.

5.2.1. INSPIRATION

Honey Badger Algorithm (HBA) mimics honey badger foraging behaviour. The honey badger either smells and digs for food or follows the honeyguide bird. The first situation is referred to as digging mode, while the second is referred to as honey mode. In the previous mode, it used its smelling skills to estimate the location of the prey; once there, it moved about the prey to find the best spot for digging and grabbing the prey. In the latter phase, the honey badger uses the honeyguide bird as a guide to locate the beehive on its own.

5.2.2 MATHEMATICAL MODEL

As previously said, HBA is divided into two phases: "digging phase" and "honey phase," which are further explained as follows:

5.3HONEY BADGER ALGORITHM

This section introduces the suggested HBA algorithm's mathematical formulation. HBA can be referred to as a global optimization method because it includes both exploration and exploitation phases.Algorithm 1 contains pseudo-code for the proposed algorithm, including population initialization, population evaluation, and parameter updates. The steps of the proposed HBA are outlined mathematically as follows. In HBA, the population of possible solutions is expressed as follows:

Population of candidate solutions =$\begin{bmatrix} x_{11}x_{12}x_{13} \dots\ x_{1D} \\ x_{21}x_{22}x_{23} \dots\ x_{2D} \\ \dots \dots \dots \dots \dots \\ x_{n1}x_{n2}x_{n3} \dots\ x_{nD} \end{bmatrix}$

*i*th position of honey badger$x_i = [x_i^1,\ \ x_i^2,\ \ \dots ,\ x_i^D]$

Step 1: Initialization phase. Based on Eq. (16), start with the number of honey badgers (population size N) and their corresponding positions:

$x_i = lb_i + r_1 \times (ub_i - lb_i), r_1$is a random number between 0 and 1

Where x_i denotes the ith honey badger position in a population of N, and lb_iandub_i denote the bottom and upper boundaries of the search zone, respectively.

Step 2: Defining intensity (I).Intensity is related to concentration strength of the prey and distance between it and I_{th} honey badger. $I_i$is smell intensity of the prey; if the smell is high, the motion will be fast and vice versa, is

given by Inverse Square Law [27] as defined by Eq(17).

$I_i = r_2 \times \frac{S}{4\pi d_i^2}, r_2$is a random number between 0 and 1 (17)

$$\mathrm{S} = (x_i - x_{i+1})^2$$

$$d_i = x_{prey} - x_i$$

where *S* is source strength or concentration strength. In Eq. (17), $d_i$denotes distance between prey and the *i*th badger.

Step 3: Update density factor. The density factor (α) controls time-varying randomization to ensure smooth transition from exploration to exploitation. Update decreasing factor α that decreases with iterations to decrease randomization with time, using Eq. (18):

$\alpha = C \times \exp\left(\frac{-t}{t_{max}}\right), t_{max} = \max \text{no. of iterations}(18)$

where *C* is a constant ≥ 1 (default = 2)

Step 4: Escaping from local optimum. This step and the two next steps are used to escape from local optima regions. In this context, the proposed algorithm uses a flag F which alters search direction for availing high opportunities for agents to scan the search-space rigorously.

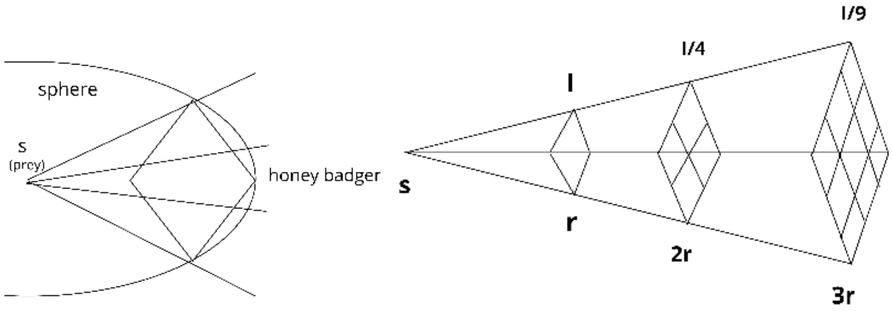


Fig.17. Inverse square law:*I* is smell intensity, *S* is location of prey, and *r* is random number between 0 and 1

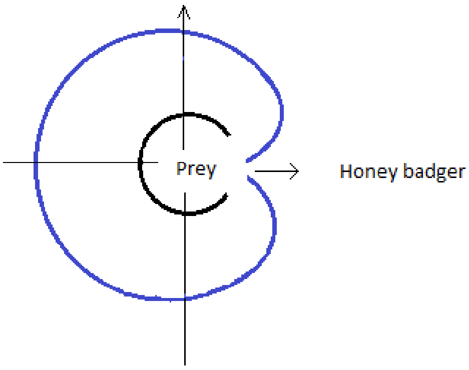


Fig.18. Digging phase: blue outline is smell intensity, black circular line shows prey location. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Step 5: Updating the agents' positions. As discussed earlier, HBA position update process (x_{new}) is divided intotwo parts which are "digging phase" and "honey phase". Following is given better explanation:

Step 5-1:Digging phase. In digging phase, a honey badger performs action similar to Cardioid shape [2] as shown

In Fig 3. The Cardioid motion can be simulated by Eq (19)

$$x_{new} = x_{prey} + F \times \beta \times I \times x_{prey} + F \times r_3 \times \alpha \times d_i \times |\cos(2\pi r_4) \times [1 - \cos 2\pi r_5]| (19)$$

where x_{prey} is position of the prey which is the best position found so far – global best position in other words. $\beta \geq 1$ (default = 6) is ability of the honey badger to get food. di is distance between prey and the *i*th honey badger, see Eq. (17). r_3, r_4 *and* $r_5$are three different random numbers between 0 and 1. F works as the flag that alters search direction, it is determined using Eq.(20):

$$F = \begin{cases} 1 & if\ r_6 \leq 0.5 \\ -1 & else, \end{cases}$$

r_6 is a random number between 0 and 1 (20)

In the digging phase, a honey badger heavily relies on smell intensity I of prey x_{prey}, distance between the badger and prey d_i, and time-varying search influence factor *α*. Moreover, during digging activity, a badger may receive any disturbance *F* which allows it to find even better prey location.

Step 5-2: Honey phase. The case when a honey badger follows honey guide bird to reach beehive can be simulated as Eq. (21):

$$x_{new} = x_{prey} + F \times r_7 \times \alpha \times d_i, r_7 \text{is a random number between 0 and } 1(21)$$

where x_{new} refer to the new position of honey badger, whereas x_{prey} is prey location, *F*and α are determined using Eqs. (20) and (18), respectively. From Eq. (21), it can be observed that a honey badger performs search close to prey location x_{prey} found so far, based on distance information d_i. At this stage, the search is influenced by search behaviour varying by time (α). Moreover, a honey badger may find disturbance *F*.

5.3.1 HONEY BADGER FLOWCHART:

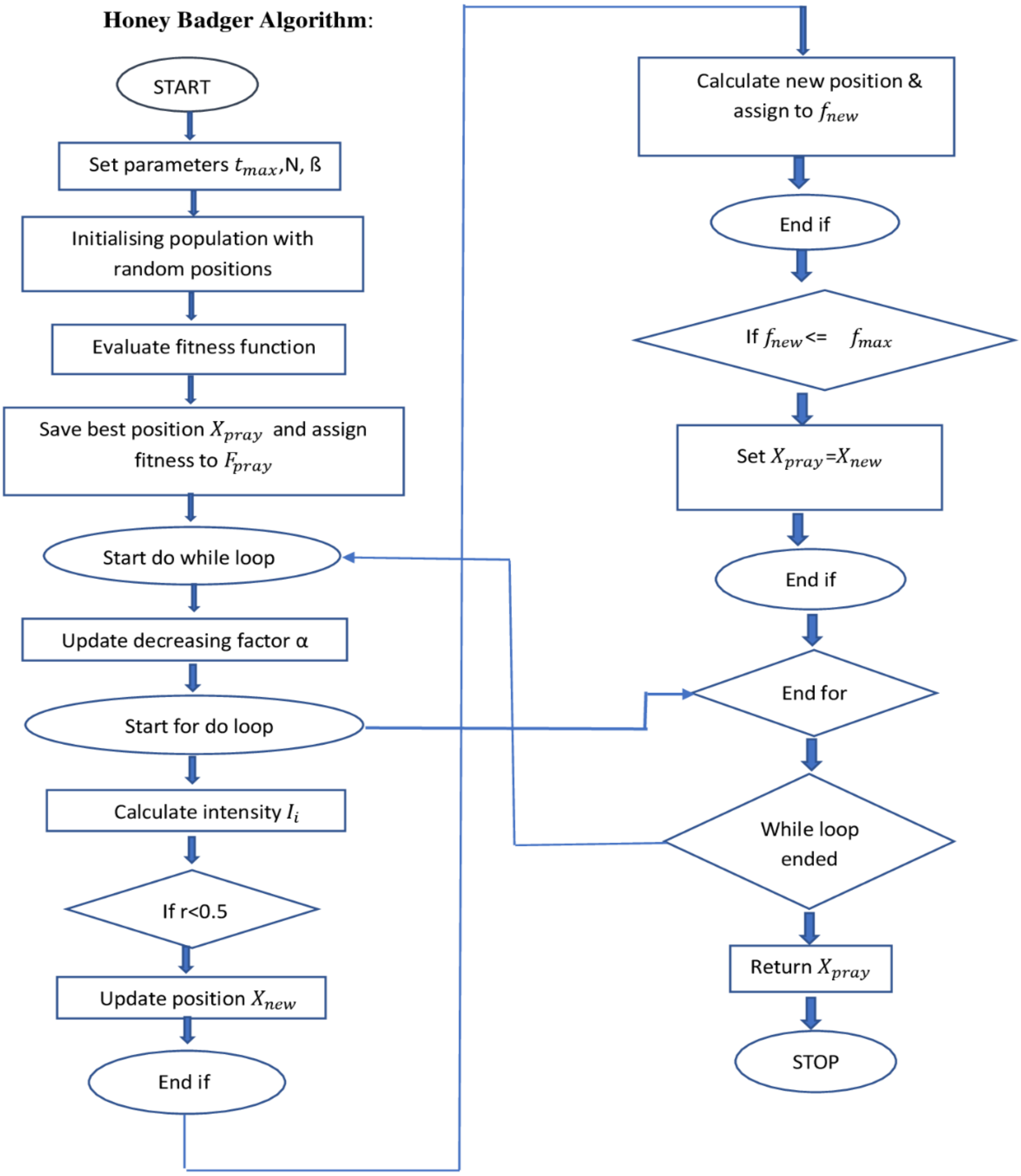

Honey Badger Algorithm:
START
Set parameters t_{max},N, ß
Initialising population with random positions
Evaluate fitness function
Save best position X_{pray} and assign fitness to F_{pray}
Start do while loop
Update decreasing factor α
Start for do loop
Calculate intensity I_i
If r<0.5
Update position X_{new}
End if
Calculate new position & assign to f_{new}
End if
If f_{new}<= f_{max}
Set X_{pray}=X_{new}
End if
End for
While loop ended
Return X_{pray}
STOP

Fig. 19 Honey Badger Algorithm Flow Chart

CHAPTER -6

EXPERIMENTAL RESULTS AND ANALYSIS:

In this section, the experimental results of CEMS-HBA over several benchmark image sets are reported. The proposed approach’s performance is compared to similar methods reported in the literature to provide strong evidence of its high accuracy and robustness. The section is divided into three subsections. First, the parameter setting of the techniques used for the comparison and the quality metrics are both described. Then, in the second subsection, the CEMS-CCO is tested using a well-known image.

6.1 EXPERIMENTAL STRUCTURE:

In the present subsection,the parameters and the metrics used for comparing different algorithms i.e. CCO and HBA is presented.

6.1.1 EXPERIMENTAL SETUP:

The main aim of experimental setup is to present comparative analysis of performance of different algorithms which are Honey Badger Algorithm(HBA) and Cluster Chaotic Optimization(CCO)[12] .The algorithm is executed for a population size of 30 and the maximum number of iterations 100.The number of thresholds taken are 4 and they are {2,3,4,5}. By employing MATLAB 9.4 on an Intel Core i5 CPU 2.7 GHz and 8 GB of RAM, all experiments were performed.[16]

6.1.2 QUALITY METRICS:

By considering the intelligent foraging behaviour of honey badger, it is very much needed to ensure that the proposed technique obtains very optimum solution when compared to the other optimization algorithms. So to ensure that, some quality metrics are used by which we can ensure that the newly proposed algorithm is a very efficient optimization algorithm over other optimization algorithms existing in the literature. They are peak signal to noise ratio (PSNR) and structural similarity index (SSIM)[16]. The PSNRcalculates the peak signal-to-noise ratio between two pictures in decibels with the use of Root mean square error as given in the equation below. This ratio is used to compare the quality of the original and compressed images. The better the quality of the compressed or reconstructed image, the higher thePSNR.

$$PSNR = 20\log(\frac{255}{RMSE}), \qquad (22)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{r} \sum_{j=1}^{c}(I(i,j)) - I_{se}(i,j)}{r \times c}},$$

(23)

Where I(I, j) is the original Gray scale image and I_{se}(i , j)is the segmented image.

Also for finding the internal structure of output segmented image, Structural similarity index measurement(SSIM)is used.

The equation of SSIM is given below

$$SSIM(I, I_{se}) = \frac{(2\mu_I \mu_{I_{se}} + \varepsilon_1)2\sigma_{I.I_{se}} + \varepsilon_2)}{(\mu_I^2 + \mu_{I_{se}}^2 + \varepsilon_1)(\sigma_I^2 + \sigma_{I_{se}}^2 + \varepsilon_2)},$$

(24)

Where $\sigma_{I.I_{se}} = \frac{1}{N-1}\sum_{i=1}^{N}(I_i + \mu_I)(I_{se,i} + \mu_{I_{se}}),$

$$\mu_I^2 + \mu_{I_t}^2 \approx 0 \; or \; \sigma_I^2 + \sigma_{I_t}^2 \approx 0$$

6.2 EXPERIMENTAL ANALYSIS ON STANDARD IMAGES:

In the initial stage of experimentation, we applied this algorithm to some standard images such as Cameraman, Lena, Peppers, and Elaine. These images are applied to the algorithm and their segmented images are computed to test the accuracy and efficiency of the algorithm. Fig.20 shows the input standard images applied to the algorithm and Fig.21 and Fig.22 shows their segmented images of CCO and HBA for different thresholds. Fig.23 shows their respective binary outputs and Fig.24 explains their ANOVA plots. It is observed that each of the input images taken is better segmented by some specific number of thresholds. The performance of the algorithm for different thresholds and their segmented outputs cannot be differentiated by normal human eyesight. The performance is best checked by using the performance checking quality metrics such as Fitness function (cross-entropy) [11], PSNR, and SSIM. Table 1 presents the quality of image segmentation for two different algorithms by applying different standard images for the different number of threshold values.

By observing Table 1 which shows the optimal thresholds, fitness values, PSNR, andSSIM, we can observe the number of thresholds to be taken for which the fitness function is better. To graphically analyze the consistency in the values ofPSNRandSSIM, the ANOVA plots are calculated. ANOVA [17] is a statistical method for comparing samples based on their means. Analysis of variance (ANOVA) is a statistical technique that is used to check if the means of two or more groups are significantly different from each other. ANOVA checks the impact of one or more factors by comparing the means of different samples. The suggested CEMS-HBA beats others in terms of the four threshold values for different standard images such as cameraman, Lena, Elaina, and peppers. It is also clearly visible that PSNRand SSIM values are very superior for the HBA algorithm when compared to the CCO algorithm.

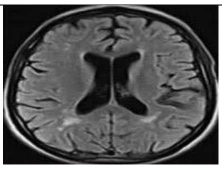
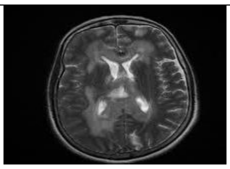
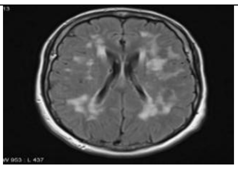
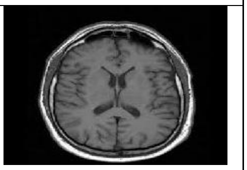
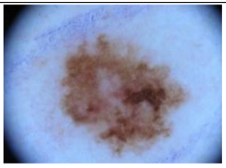
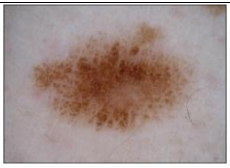
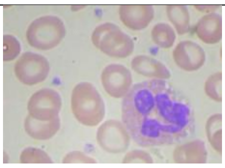
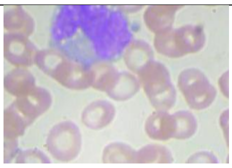
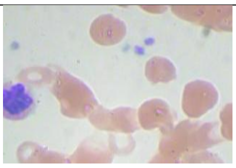
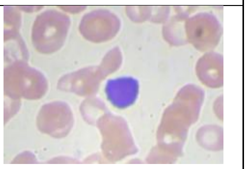
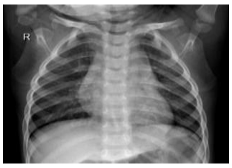
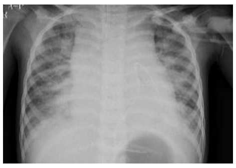
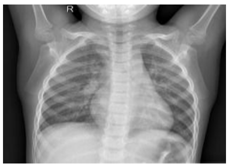
Standard Images				
Brain Tumor Images				
Skin cancer Images				
Blood cell Images				
Chest X-ray Images				

Fig. 20Input images taken for segmentation.

Table 1 Experimental results show the comparison between the best fitness values and performance parameters i.e., PSNRand SSIMof CEMS-CCO and CEMS-HBA.

Algorithms			**CEMS-CCO**				**CEMS-HBA**			
S.no	Name of the image	No. of Threshold	Optimal Thresholds	Best Fitness values	PSNR	SSIM	Optimal Thresholds	Best Fitness values	PSNR	SSIM
1	Peppers	2	38,115	1.8869	21.2953	0.6825	38,115	1.8869	21.2953	0.6825
		3	27,68,124	1.2085	22.7104	0.7053	27,68,124	1.2080	22.7104	0.7053
		4	17,39,74,129	0.7228	23.0434	0.7189	17,38,75,126	0.7175	23.0752	0.7210
		5	16,38,76,110,152	0.5352	26.1930	0.7450	17,37,71,111,157	0.5320	26.2026	0.7453
2	Elaine	2	91,148	1.3025	22.2128	0.5998	91,148	1.3025	22.2128	0.5998
		3	7,61,16,163	0.7864	24.2631	0.6581	76,115,162	0.7855	24.2631	0.6581
		4	67,101,137,176	0.5309	26.1796	0.7036	69,101,135,175	0.5134	26.1430	0.7039
		5	64,88,113,139,176	0.3667	27.6897	0.7363	63,89,101,147,184	0.3584	27.6145	0.7425
3	Cameraman	2	23,82	1.4305	20.8081	0.7605	23,82	0.7968	20.8081	0.7605
		3	22,68,137	0.0000	25.0675	0.8413	22,68,138	0.0000	25.0675	0.8413
		4	13,28,72,139	0.0000	25.1470	0.8480	12,25,71,138	0.0000	25.2213	0.8481
		5	12,20,49,91,141	0.0000	26.0517	0.8658	12,21,45,87,141	0.0000	25.9202	0.8610
4	Lena	2	58,114	1.5939	21.8741	0.7514	58,114	1.5939	21.8741	0.7514
		3	52,94,142	0.7964	25.2524	0.7885	52,94,142	0.7925	25.2524	0.7885
		4	45,69,102,146	0.5138	26.4603	0.8118	45,70,103,146	0.5113	26.4195	0.8112
		5	37,56,86,116,151	0.3769	28.6766	0.8304	43,65,92,120,156	0.3653	28.5254	0.8308
5	Brain MRI (1)	2	19,60	2.1344	21.7692	0.6943	19,60	2.1344	21.7692	0.6943
		3	8,21,64	0.8161	21.8603	0.7030	18,51,106	1.2359	24.2868	0.8040
		4	8,20,54,107	0.9141	24.4439	0.8123	8,20,53,107	0.9105	24.5082	0.8146
		5	8,18,37,69,114	0.6887	26.1201	0.8417	7,17,36,70,115	0.6743	25.8299	0.8423
6	Brain MRI (2)	2	8,66	1.6291	23.2125	0.7595	8,66	1.6535	23.2125	0.7595
		3	2,9,67	1.0244	23.4986	0.7838	5,24,73	0.7704	24.3244	0.8077
		4	2,6,27,74	0.6620	24.5328	0.8212	2,6,27,74	0.5227	24.4616	0.8220
		5	2,6,24,62,106	0.4868	27.0333	0.8687	2,6,25,62,105	0.3779	27.2476	0.8693
7	Brain MRI(3)	2	23,72	1.6535	20.6682	0.7930	23,72	1.6535	20.6682	0.7930
		3	22,65,131	0.7704	24.9680	0.8287	22,65,131	0.7702	25.2690	0.8301
		4	19,45,80,132	0.5240	25.6243	0.8492	19,41,76,133	0.5227	25.9719	0.8495
		5	19,41,70,109,154	0.3872	28.3167	0.8736	19,40,73,112,157	0.3779	28.3721	0.8735
8	Brain MRI(4)	2	6,59	2.6779	20.0915	0.7480	6,59	2.6799	20.0915	0.7480
		3	3,18,78	1.3745	20.9228	0.8061	3,18,78	1.3711	20.9228	0.8061
		4	3,16,66,145	0.9762	25.7820	0.8380	3,16,66,140	0.9740	25.6905	0.8378
		5	3,12,39,79,148	0.6225	26.9181	0.8674	3,11,38,80,145	0.6179	27.1116	0.8671
9	Chest X-ray (1)	2	39,96	1.7288	21.2583	0.8007	39,96	1.7288	21.2583	0.8007
		3	35,73,122	1.0064	24.3774	0.7978	35,73,122	1.0038	24.3672	0.7980
		4	34,63,95,141	0.6208	27.1919	0.8088	34,64,95,139	0.6200	26.9398	0.8096
		5	16,38,65,96,137	0.4771	27.1581	0.8102	13,36,64,95,139	0.4485	27.0379	0.8124
10	Chest X-ray (2)	2	47,117	3.5759	20.6373	0.6272	14,86	3.5759	18.8559	0.6811
		3	34,71,124	1.9723	22.6954	0.6498	12,56,128	1.9723	22.7404	0.6823
		4	31,59,95,141	1.3323	24.3273	0.6547	11,43,90,144	1.3216	25.1911	0.6868
		5	31,51,77,103,157	0.9042	26.6179	0.6633	10,31,63,106,152	0.9279	26.7574	0.7013

11	Chest X-ray (3)	2	14,86	2.6279	18.8559	0.6811	47,107	2.6279	20.6373	0.6272
		3	12,57,130	1.3673	22.9116	0.6803	34,70,124	1.3671	22.6868	0.6491
		4	11,40,84,143	0.8515	25.1370	0.6875	31,59,95,140	0.8483	24.6046	0.6507
		5	11,32,67,117,152	0.6322	26.2729	0.6964	29,52,80,115,157	0.6229	26.5606	0.6647
12	Chest X-ray (4)	2	39,129	2.1948	20.3760	0.7821	39,129	2.1948	20.3760	0.7821
		3	16,47,132	1.2562	20.6413	0.7845	15,46,131	1.2562	20.6602	0.7844
		4	14,40,95,152	0.8747	23.6206	0.7785	14,39,96,153	0.8718	23.5146	0.7779
		5	14,35,81,123,173	0.6831	26.0603	0.7857	14,36,79,121,167	0.6961	25.8150	0.7854
13	Skin Cancer (1)	2	27,129	1.1957	22.6552	0.8437	26,129	1.1957	22.6552	0.8437
		3	19,78,137	0.7650	24.2245	0.8466	19,77,137	0.7650	24.2954	0.8464
		4	18,67,111,152	0.4919	26.3445	0.8325	18,69,113,152	0.4892	26.3759	0.8335
		5	12,36,77,117,160	0.3693	26.6375	0.8334	11,33,75,116,155	0.3362	26.7008	0.8328
14	Skin Cancer (2)	2	54,130	0.7384	23.8895	0.9127	54,130	0.7384	23.8895	0.9127
		3	42,78,157	0.4537	26.0437	0.9134	42,78,157	0.4537	25.9714	0.9128
		4	36,57,99,167	0.3185	27.0147	0.9174	37,57,99,172	0.3159	27.2186	0.9168
		5	35,50,74,119,190	0.2346	29.1642	0.9140	35,51,80,130,191	0.2338	28.7757	0.9170
15	Skin Cancer (3)	2	6,135	0.3504	27.9269	0.9472	6,135	0.3504	27.9269	0.9472
		3	18,120,144	0.1979	30.0912	0.9484	6,121,145	0.1979	30.1313	0.9460
		4	8,121,146,230	0.1220	32.8502	0.9490	6,121,145,197	0.1212	32.8217	0.9485
		5	1,49,120,146,186	0.0752	34.8428	0.9416	61,80,138,153,232	0.0745	35.0183	0.9406
16	Skin Cancer (4)	2	7,123	0.7766	24.9421	0.8958	7,122	0.7767	24.9421	0.8958
		3	7,97,137	0.4217	27.3176	0.8934	7,96,136	0.4216	27.3176	0.8934
		4	14,88,116,147	0.2789	28.8498	0.8885	6,87,117,147	0.2783	28.9771	0.8903
		5	21,79,103,126,152	0.2046	30.3035	0.8810	6,83,109,135,157	0.2118	30.3868	0.8809
17	Neutrophil Blood Cell	2	149,181	0.1511	30.7216	0.8622	149,181	0.1511	30.7216	0.8622
		3	143,173,191	0.0947	32.7446	0.8923	145,172,189	0.0947	32.7446	0.8923
		4	143,165,179,194	0.0663	34.5990	0.9026	25,145,172,189	0.0947	32.7446	0.8923
		5	129,146,169,183,198	0.0509	35.9991	0.9166	140,161,173,186,198	0.0451	35.9756	0.9223
18	Monocyte Blood Cell	2	155,193	0.2151	28.9800	0.8218	155,193	0.2151	28.9800	0.8218
		3	151,179,201	0.1205	31.5673	0.8635	152,179,201	0.1205	31.4968	0.8618
		4	150,176,190,205	0.0830	33.0391	0.8902	150,175,183,211	0.0813	33.3167	0.8776
		5	145,166,184,201,216	0.0696	33.8844	0.8931	139,158,177,195,212	0.0595	34.1693	0.8851
19	Lymphocyte Blood Cell	2	146,196	0.2094	28.9515	0.8461	146,196	0.2094	28.9515	0.8461
		3	141,180,203	0.1073	31.8056	0.8823	143,180,204	0.1073	31.8056	0.8823
		4	143,179,198,217	0.0730	33.5666	0.8927	142,179,199,217	0.0724	33.6367	0.8884
		5	147,174,192,205,218	0.0614	34.5023	0.9139	141,175,189,205,220	0.0489	35.3997	0.9150
20	Esinophil Blood Cell	2	141,193	0.1749	29.8101	0.8787	141,193	0.1749	29.8101	0.8787
		3	137,179,201	0.1032	32.0259	0.9030	139,179,201	0.1032	32.0259	0.9030
		4	135,177,195,213	0.0744	34.2518	0.9067	138,176,196,212	0.0630	34.2900	0.9089
		5	143,172,180,195,211	0.0512	35.0475	0.9200	136,170,183,199,213	0.0460	35.5832	0.9251

6.3 EXPERIMENTAL RESULTS ON MEDICAL IMAGES:

The proposed CCO algorithm is experimented over a variety of medical images and results are generated. Later a comparative analysis is made with HBA as CCO is already considered the best algorithm by comparing with HHO, PSO, DE, HS, ABC algorithms. So, we opted CCO and now from our results it is obtained that HBA is giving much better results than CCO. To demonstrate a comparative analysis between CCO and HBA different medical images which are extensively used in literature are taken. The results are organized into four sections for this reason: multilevel segmentation for brain tumour, skin cancer, blood cell and chest X-ray datasets.

6.3.1 MULTI-LEVEL SEGMENTATION FOR BRAIN TUMOUR IMAGES:

We have taken four sets of brain tumour images to study and analyse the behaviour of both CCO and HBA optimization algorithms. The images that we opted for are shown in Fig.20 We have performed segmentation with several thresholds($n = 2,3,4,5$) respectively. Fig.21 shows their respective CCO segmented outputs. Fig.22 shows the output of the input dataset images of the brain tumour images after segmentation using the HBA algorithm with the respective threshold values. Fig.23 shows their corresponding binary outputs. Proper segmentation is required for differentiating the lesion textures and disease progression in the images. The input data set is taken from the Kaggle website [18]

The respective optimal threshold values are listed in the table 1 along with their best fitness values. The less fitness value indicates better segmentation. In most of the cases, we see that CEMS-HBA gives better fitness values when compared to other algorithms. Once the best fitness values are determined the performance parameters PSNR and SSIM and ANOVA plots are evaluated and shown in Fig.24 for a different number of thresholds such as ($n = 2,3,4,5$). In the first set, we can see that for threshold 2 both performed the same and HBA showed better results for thresholds 4, and 5 while CCO showed less deviation for threshold 3. For the second, third, and fourth datasets taken we can see HBA gave the best results or either both gave the same result. But in most cases, we can see that HBA gives better results for brain tumor diagnosis.

6.3.2 MULTI-LEVEL SEGMENTATION FOR SKIN CANCER:

Here we have chosen four sets of skin cancer images to check the performance of both CEMS-CCO and CEMS-HBA algorithms. In this, we aim to identify the region of interest. In Fig.20 the input skin cancer images are shown and in Fig.21 shows the CCO segmented images. Fig.22 represents the HBA segmented outputs. Fig.23 shows their corresponding binary outputs. The dataset that is used here for the study is taken from the Kaggle website [20]. The Table 1 contains the optimal threshold values obtained using the minimum cross-entropy criteria calculated from the combination of thresholds used in benchmark images $= 2,3,4,5$. Later the PSNR and SSIM performance parameters are also evaluated and the ANOVA plots as shown in Fig.24 that examine the deviations are plotted. Here in most of the cases, we can see that HBA has given less deviation when compared with CCO but in some cases where the number of thresholds is taken as 2, they gave the same deviation which proves that HBA is much better than CCO.

6.3.3 MULTI-LEVEL SEGMENTATION FOR BLOOD CELL:

The next step is to examine the results of the blood cells. Here we have taken four data sets namely the neutrophil, monocyte, lymphocyte, and the eosinophil blood cells from the following dataset [21]. Also, the same number of threshold values are considered i.e., $n = 2,3,4,5$ from the dataset shown in Table 1. The same process is carried out here again to make a comparative analysis between the CEMS-CCO and CEMS-HBA algorithms. The Table1 shows the optimal threshold values obtained. It is evident from the values that as the number of threshold values increases a better result in terms of best fitness values is obtained. After image segmentation the performance parameters i.e., the PSNR and SSIM are evaluated to compute the image performance. In Fig.20 the input skin cancer images are shown and in Fig.21 the CCO segmented outputs. The Fig.22 shows the HBA segmented images and Fig.23 shows the corresponding binary images thus representing their region of interest. Both CCO and HBA obtained the same results for $n = 2$ and 3. But for $n = 4$ and 5 we can see that HBA gave better values. The ANOVA plots are plotted for a clear analysis in Fig.24.

6.3.4 MULTI-LEVEL SEGMENTATION FOR CHEST X-RAY:

The final step is to process the chest X-ray images that are obtained Kaggle website [19] which contains different diseases that require an accurate diagnosis. Fig.20 represents the images that have been selected from the dataset and Fig.21 and Fig.22shows their respective

CCO and HBA segmented images. Fig.23 shows their corresponding binary outputs. In chest X-ray images it is difficult to spot the exact region of interest. The correct positioning of the threshold helps in the early diagnosis of the disease. Therefore, we find CCO and HBA very helpful for multilevel segmentation owing to their better performance and accuracy for complex images. The table shows the list of optimal threshold values along with the performance parameters namely the PSNR and SSIM. In the table, it is visible that both gave the same values for $n = 2$. In the first data set, we see that HBA is better than CCO but in the second and fourth dataset for $n = 5$ we see slightly better performance of CCO than HBA. But in all the other cases we considerably see that there is less deviation for HBA than CCO in the ANOVA plots plotted in Fig.24.

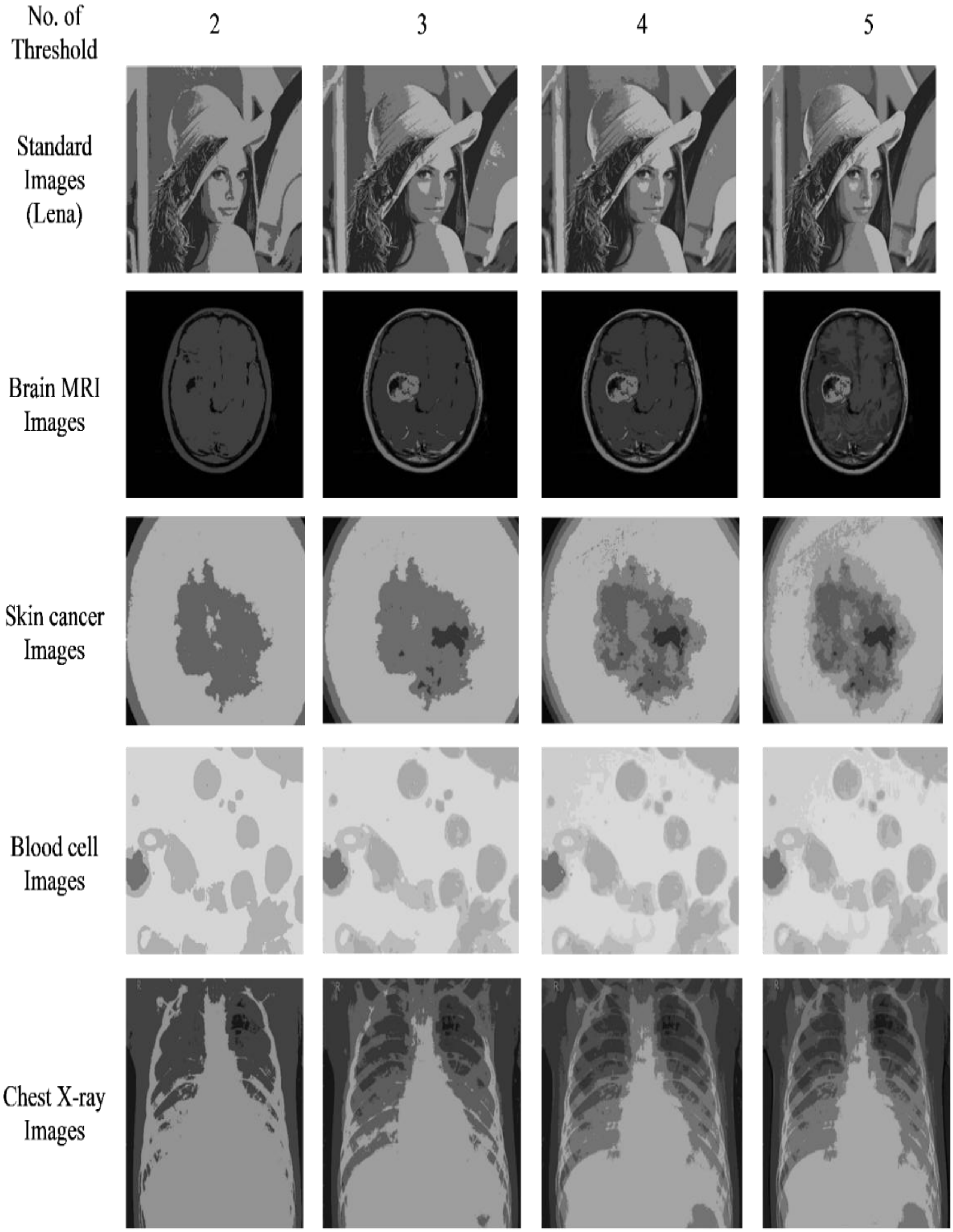


Fig. 21Output images after segmentation using CCO algorithm.

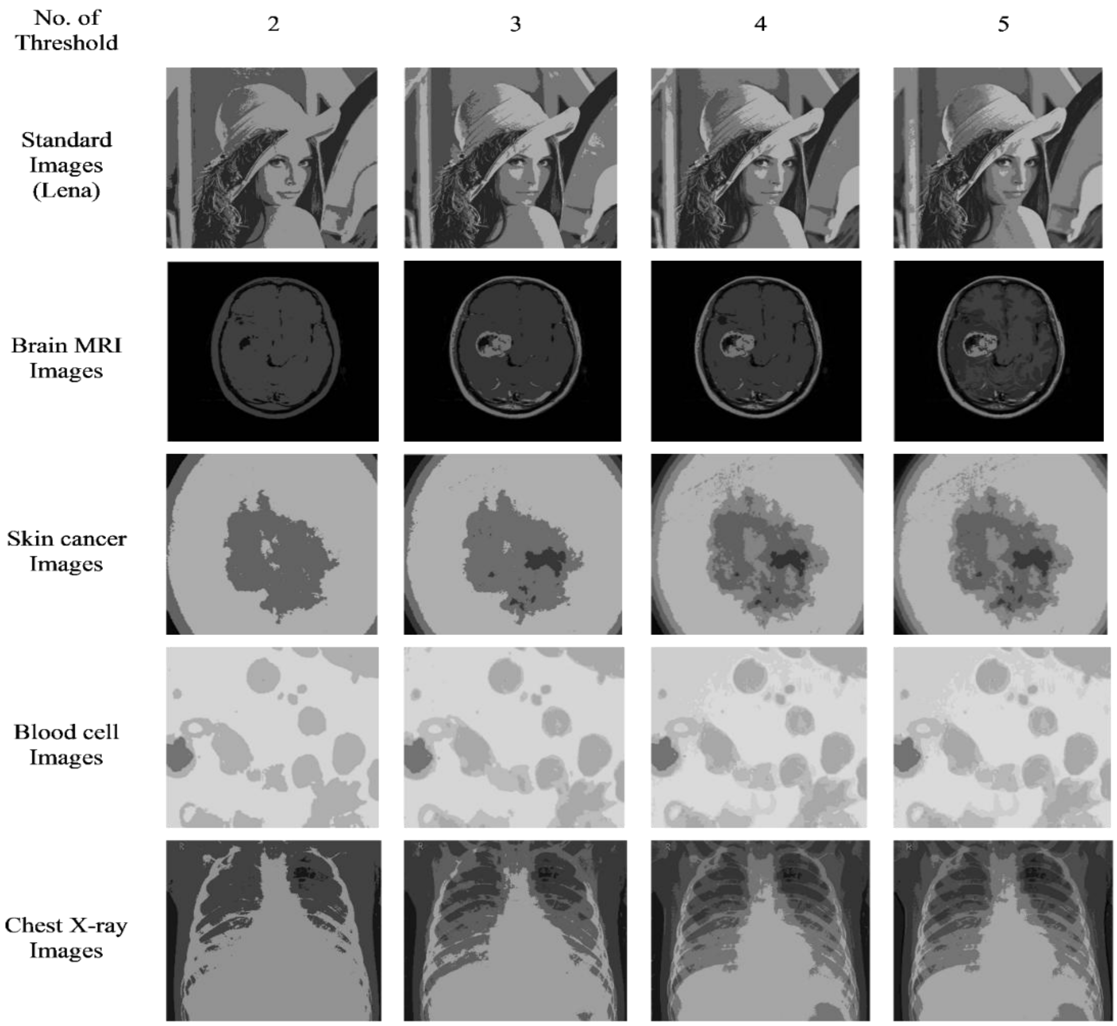


Fig.22Output images after segmentation using HBA algorithm.

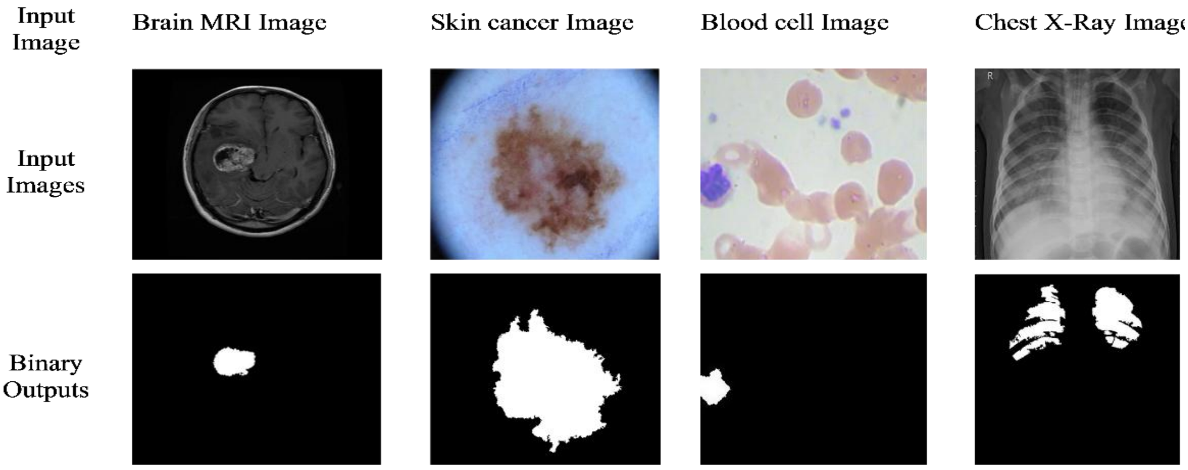


Fig.23 Output binary images after segmentation

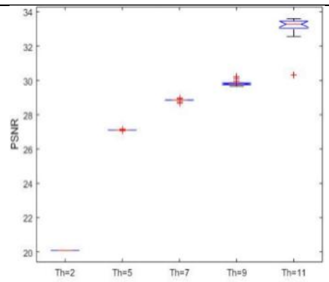
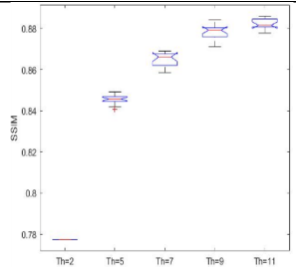
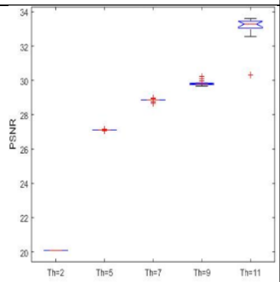
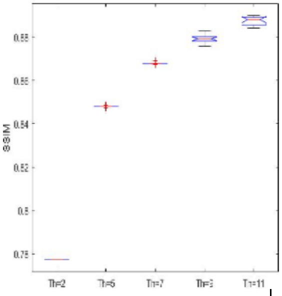
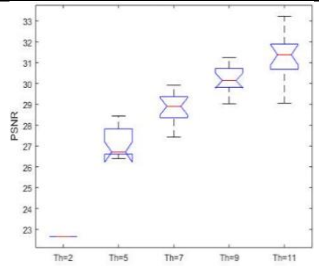
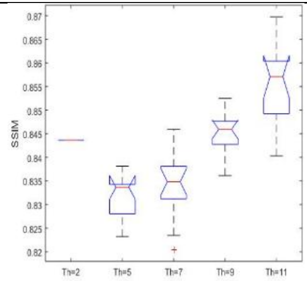
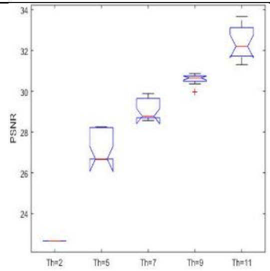
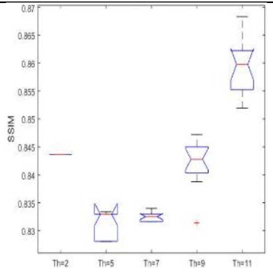
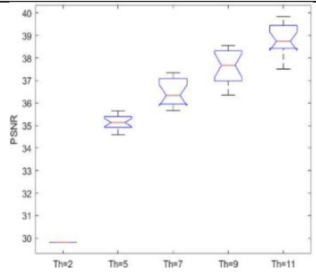
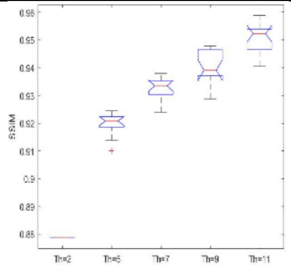
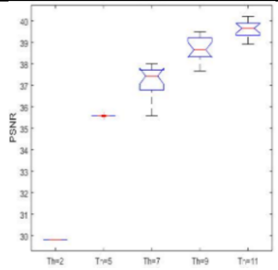
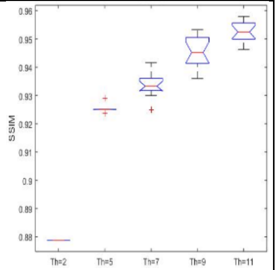
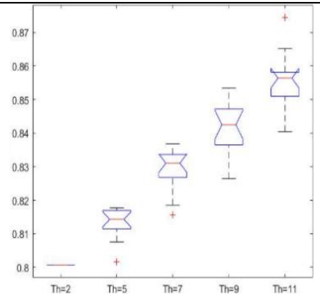
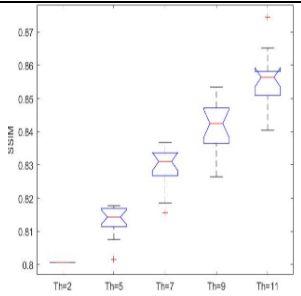
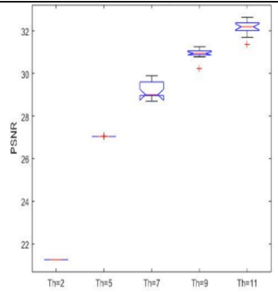
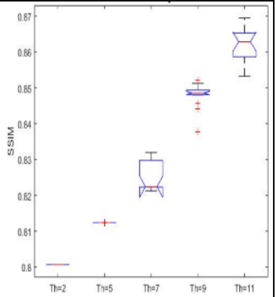
	CCO		HBA	
	PSNR	SSIM	PSNR	SSIM
Brain MRI Image				
Skin cancer Image				
Blood cell Image				
Chest X-ray Image				

Fig.24 ANOVA plots representing performance comparison between CCO and HBA algorithm

CHAPTER -7

CONCLUSION:

The Honey Badger Algorithm is one of the best approach as far as our results. We have made a comparative analysis of both HBA and CCO which is one of the novel approach when compared to many other algorithms. Both CCO and HBA uses minimum cross entropy criteria. To compare the two outputs of HBA and CCO, we used different Performance parameters which are used to check the similarity between the segmented and the original images for different threshold combinations. PSNR(peak signal to noise ratio) and SSIM(structural similarity index measure).

PSNRis used to determine the image segmentation quality i.e., high PSNRvalue means better image segmentation quality and SSIMis used to measure the similarity between the images. Now we plot the outputs with respect to their SSIMand PSNRvalues using the keyword ‘ANOVA’ in MATLAB, where ANOVA means analysis of variance which is a statistical method to differentiate how much variance occurs among the population. To determine the above parameters we have used many images and applied the algorithms to get the ANOVA plots. After analysing the above ANOVA plots we can clearly say that the HBA outputs has very less variation among the population with respect to both SSIMandPSNR

APPENDIX

%main code

```
clc
clear all
close all
tic
za=rand;
zb=rand;
alpha=rand;
N=30;
M=5;
I=imread('elaine.gif');
[r1,c1,s1]=size(I);
if s1>1
    I=rgb2gray(I);
end
a=0;
b=255;
itrmx=100;
for i=1:N
    r(i,:)=a+rand(1,M)*(b-a);
end
for i=1:N
        T=sort(round(r(i,:)),2);
        fitness(i)=CEP(I,T);
end
[vx,ix]=sort(fitness);
G=vx(1);
```

```
CS=6;
P1=r(1:CS,:);
fitness1=fitness(1:CS);

P2=r(CS+1:2*CS,:);
fitness2=r(CS+1:2*CS);

P3=r(2*CS+1:3*CS,:);
fitness3=fitness(2*CS+1:3*CS);

P4=r(3*CS+1:4*CS,:);
fitness4=fitness(3*CS+1:4*CS);

P5=r(4*CS+1:5*CS,:);
fitness5=fitness(4*CS+1:5*CS);

itrmx=100;
for i=1:itrmx
[Bestsol1,fitness1,P1, bestfitness1,G1]=CCO12(I,P1,fitness1,za,zb,alpha);
[Bestsol2,fitness2,P2, bestfitness2,G2]=CCO12(I,P2,fitness2,za,zb,alpha);
[Bestsol3,fitness3,P3, bestfitness3,G3]=CCO12(I,P3,fitness3,za,zb,alpha);
[Bestsol4,fitness4,P4, bestfitness4,G4]=CCO12(I,P4,fitness4,za,zb,alpha);
[Bestsol5,fitness5,P5, bestfitness5,G5]=CCO12(I,P5,fitness5,za,zb,alpha);
bestfitness4;
BS=[Bestsol1;Bestsol2;Bestsol3;Bestsol4;Bestsol5];
BF=[bestfitness1,bestfitness2,bestfitness3,bestfitness4,bestfitness5];
[Vx,Ix]=sort(BF);
GB=BS(Ix(1),:);
BF=Vx(1);
R=0+rand*((2*pi)-0);
```

```
cbest1=(Bestsol1+(GB+Bestsol1)*rand*cos(alpha*R));

R=0+rand*((2*pi)-0);

cbest2=(Bestsol2+(GB+Bestsol2)*rand*cos(alpha*R));

R=0+rand*((2*pi)-0);

cbest3=(Bestsol3+(GB+Bestsol3)*rand*cos(alpha*R));

R=0+rand*((2*pi)-0);

cbest4=(Bestsol4+(GB+Bestsol4)*rand*cos(alpha*R));

R=0+rand*((2*pi)-0);

cbest5=(Bestsol5+(GB+Bestsol5)*rand*cos(alpha*R));

f=find(cbest1<0);

cbest1(f)=0;

f=find(cbest1>255);

cbest1(f)=255;

f=find(cbest2<0);

cbest2(f)=0;

f=find(cbest2>255);

cbest1(f)=255;

f=find(cbest3<0);

cbest3(f)=0;

f=find(cbest3>255);

cbest3(f)=255;

f=find(cbest4<0);

cbest4(f)=0;

f=find(cbest4>255);

cbest4(f)=255;

f=find(cbest5<0);

cbest5(f)=0;

f=find(cbest5>255);

cbest5(f)=255;

G=[G, BF];
```

```
end
sort(GB);
Th=round(sort(GB));
J1=double(I);
[PSNR,J2]=psnrfn(J1,Th);

J2=uint8(J2);
imshow(J2);
%figure;
% imshow(I);
% figure;
% imhist(J2);
PSNRx=(psnr(I,J2))
SSIMx=(ssim(I,J2))
toc
```

%cco

```
function [Bestsol,fitness,P, bestfitness,G]=CCO12(I,P,fitness,za,zb,alpha)
N=size(P,1);
M=size(P,2);
    [v,ix]=sort(fitness);
    BL=ix(1);
    Bestsol=P(BL,:);
    bestfitness=v(1);
    G=bestfitness;
for itr=1:1
        q=abs(bestfitness)/N;
for i=1:N
for j=1:M
                z=-1+rand*(1-(-1));
                P1(i,j)=P(i,j)+(0.4*rand*((Bestsol(j))-P(i,j)));
if (P1(i,j)<0)
                    P1(i,j)=0;
elseif (P1(i,j)>255)
                    P1(i,j)=255;
end
end
end

for i=1:N
        T=sort(round(P1(i,:)),2);
        fitness1(i)=CEP(I,T);
end
        hA=[];
        hB=[];

for i=1:N
for j=1:M
                za=rand;
                zb=rand;
                R=0+rand*((2*pi)-0);
                hA(i,j)=P1(i,j)+(P1(i,j)*za*cos(alpha*R));
                R=0+rand*((2*pi)-0);
                hB(i,j)=P1(i,j)-(P1(i,j)*zb*cos(alpha*R));

end
end
        f=find(hA<0);
        hA(f)=0;
        f=find(hA>255);
        hA(f)=255;
        f=find(hB<0);
        hB(f)=0;
        f=find(hB>255);
        hB(f)=255;
for i=1:N
            T1=sort(round(hA(i,:)),2);
            T2=sort(round(hB(i,:)),2);
            fitness2(i)=CEP(I,T1);
            fitness3(i)=CEP(I,T2);
end

for i=1:N
if (fitness2(i)<fitness1(i))
                    U(i,:)=[hA(i,:)];
                    fitness4(i)=fitness2(i);

elseif (fitness3(i)<fitness1(i))
                U(i,:)=[hB(i,:)];
                fitness4(i)=fitness3(i);
else
                U(i,:)=P1(i,:);
                fitness4(i)=fitness1(i);
end
end

for i=1:N
if fitness4(i)<fitness(i)
               P(i,:)=U(i,:);
               fitness(i)=fitness4(i);
```

```
end
end
       [v,ix]=sort(fitness);
        BL=ix(1);
        Bestsol=P(BL,:);
        bestfitness=v(1);
        G=[G,bestfitness];
        fitness1=[];
        fitness2=[];
        fitness3=[];
end
end
```

% cco multi thresholding

```
function[PSNRx,SSIMx]=ccomlt(I,M)
za=rand;
zb=rand;
alpha=rand;
N=30;
[r1,c1,s1]=size(I);
if s1>1
    I=rgb2gray(I);
end
a=0;
b=255;
itrmx=100;
for i=1:N
    r(i,:)=a+rand(1,M)*(b-a);
end
for i=1:N
```

```
        T=sort(round(r(i,:)),2);
        fitness(i)=CEP(I,T);
end
[vx,ix]=sort(fitness);
G=vx(1);
CS=6;
P1=r(1:CS,:);
fitness1=fitness(1:CS);

P2=r(CS+1:2*CS,:);
fitness2=r(CS+1:2*CS);

P3=r(2*CS+1:3*CS,:);
fitness3=fitness(2*CS+1:3*CS);

P4=r(3*CS+1:4*CS,:);
fitness4=fitness(3*CS+1:4*CS);

P5=r(4*CS+1:5*CS,:);
fitness5=fitness(4*CS+1:5*CS);

itrmx=100;
for i=1:itrmx
[Bestsol1,fitness1,P1, bestfitness1,G1]=CCO12(I,P1,fitness1,za,zb,alpha);
[Bestsol2,fitness2,P2, bestfitness2,G2]=CCO12(I,P2,fitness2,za,zb,alpha);
[Bestsol3,fitness3,P3, bestfitness3,G3]=CCO12(I,P3,fitness3,za,zb,alpha);
[Bestsol4,fitness4,P4, bestfitness4,G4]=CCO12(I,P4,fitness4,za,zb,alpha);
[Bestsol5,fitness5,P5, bestfitness5,G5]=CCO12(I,P5,fitness5,za,zb,alpha);
bestfitness4;
BS=[Bestsol1;Bestsol2;Bestsol3;Bestsol4;Bestsol5];
```

```
BF=[bestfitness1,bestfitness2,bestfitness3,bestfitness4,bestfitness5];

[Vx,Ix]=sort(BF);

GB=BS(Ix(1),:);

BF=Vx(1);

R=0+rand*((2*pi)-0);

cbest1=(Bestsol1+(GB+Bestsol1)*rand*cos(alpha*R));

R=0+rand*((2*pi)-0);

cbest2=(Bestsol2+(GB+Bestsol2)*rand*cos(alpha*R));

R=0+rand*((2*pi)-0);

cbest3=(Bestsol3+(GB+Bestsol3)*rand*cos(alpha*R));

R=0+rand*((2*pi)-0);

cbest4=(Bestsol4+(GB+Bestsol4)*rand*cos(alpha*R));

R=0+rand*((2*pi)-0);

cbest5=(Bestsol5+(GB+Bestsol5)*rand*cos(alpha*R));

f=find(cbest1<0);

cbest1(f)=0;

f=find(cbest1>255);

cbest1(f)=255;

f=find(cbest2<0);

cbest2(f)=0;

f=find(cbest2>255);

cbest1(f)=255;

f=find(cbest3<0);

cbest3(f)=0;

f=find(cbest3>255);

cbest3(f)=255;

f=find(cbest4<0);

cbest4(f)=0;

f=find(cbest4>255);

cbest4(f)=255;
```

```
f=find(cbest5<0);
cbest5(f)=0;
f=find(cbest5>255);
cbest5(f)=255;
G=[G, BF];
end
sort(GB);
Th=round(sort(GB));
J1=double(I);
[PSNR,J2]=psnrfn(J1,Th);

J2=uint8(J2);
imshow(J2);
disp(Th);
% figure;
% imshow(I);
% figure;
% imhist(J2);
PSNRx=(psnr(I,J2));
SSIMx=(ssim(I,J2));
end
```

% CEP

```
function Y=CEP(I,T)
[C,V]=imhist(I);
L=length(T)+1;%no.of regions
k=0;
m=1;
for i=1:L%loop repeats for T+1 regions
if i~=L
```

```
        f1=find((V>=k)&(V<=T(i)));
if ((numel(f1)>0)&& (sum(C(f1))>0))
            f=max(f1);
            p=C(m:f)/sum(C);
            mu=sum(p.*V(m:f))/sum(p);%2nd equation
            A=(lognew(V(m:f),mu));
            H(i)=sum(V(m:f).*(p).*A');
            k=T(i)+1;
            m=f+1;
else
            H(i)=0;
            k=T(i)+1;
end

else
        f1=find((V>=k)&(V<=255));
if ((numel(f1)>0)&&(sum(C(f1))>0))
            f=max(f1);
            p=C(m:f)/sum(C);
            mu=sum(p.*V(m:f))/sum(p);
            A=(lognew(V(m:f),mu));
            H(i)=sum(V(m:f).*(p).*A');
else
            H(i)=0;
end
end
end
Y=sum(H);
end
```

% CEP4

```
I=imread('C:\Users\Lenovo\Downloads\lena.jpg');
T=[10,122,220];
[C,V]=imhist(I);
L=length(T)+1;
k=0;
m=1;
for i=1:L
if i~=L
        f1=find((V>=k)&(V<=T(i)));
if ((numel(f1)>0)&& (sum(C(f1))>0))
            f=max(f1);
            p=V(m:f);
            A1=sum(p)/(max(f1));
            D1=abs(p-A1);
            sumd=sum(exp(D1));
            SD1=exp(D1)/sumd;
            E1=-sum(SD1.*log(SD1));
            k=T(i)+1;
            m=f+1;
else
            E1=0;
            k=T(i)+1;
end

else
        f1=find((V>=k)&(V<=255));
if ((numel(f1)>0)&&(sum(C(f1))>0))
            f=max(f1);
            p=V(m:f);
            A1=sum(p)/(max(f1));
            D1=abs(p-A1);
            sumd=sum(exp(D1));
            SD1=exp(D1)/sumd;
            E1=-sum(SD1.*log(SD1));
else
            E1=0;
end
end
    Y=sum(E1);
end
disp(Y);
```

% honey badger code

```
function [Xprey, Food_Score,CNVG] = HBA(objfunc, dim,lb,ub,tmax,N,II)
beta       = 6;     % the ability of HB to get the food  Eq.(4)
C       = 2;     %constant in Eq. (3)
vec_flag=[1,-1];
%initialization
X=initialization(N,dim,ub,lb);
%Evaluation
fitness = fun_calcobjfunc(objfunc,II, X);
[GYbest, gbest] = min(fitness);
Xprey = X(gbest,:);
for t = 1:tmax
    alpha=C*exp(-t/tmax);   %density factor in Eq. (3)
    I=Intensity(N,Xprey,X); %intensity in Eq. (2)
for i=1:N
        r =rand();
        F=vec_flag(floor(2*rand()+1));
for j=1:1:dim
            di=((Xprey(j)-X(i,j)));
if r<.5
                r3=rand;                r4=rand;                r5=rand;


                Xnew(i,j)=Xprey(j) +F*beta*I(i)*
Xprey(j)+F*r3*alpha*(di)*abs(cos(2*pi*r4)*(1-cos(2*pi*r5)));
else
                r7=rand;
                Xnew(i,j)=Xprey(j)+F*r7*alpha*di;
end
end

FU=Xnew(i,:)>ub;FL=Xnew(i,:)<lb;Xnew(i,:)=(Xnew(i,:).*(~(FU+FL)))+ub.*FU+lb
.*FL;


        tempFitness = fun_calcobjfunc(objfunc,II, Xnew(i,:));
if tempFitness<fitness(i)
            fitness(i)=tempFitness;
            X(i,:)= Xnew(i,:);
end
end
    FU=X>ub;FL=X<lb;X=(X.*(~(FU+FL)))+ub.*FU+lb.*FL;
    [Ybest,index] = min(fitness);
    CNVG(t)=min(Ybest);
if Ybest<GYbest
        GYbest=Ybest;
        Xprey = X(index,:);
end
end
Food_Score = GYbest;
```

```
end


function Y = fun_calcobjfunc(func,II, X)

N = size(X,1);

for i = 1:N

    Y(i) = func(II,sort(round(X(i,:))));

end

end

function I=Intensity(N,Xprey,X)

for i=1:N-1

    di(i) =( norm((X(i,:)-Xprey+eps))).^2;

    S(i)=( norm((X(i,:)-X(i+1,:)+eps))).^2;

end

di(N)=( norm((X(N,:)-Xprey+eps))).^2;

S(N)=( norm((X(N,:)-X(1,:)+eps))).^2;

for i=1:N

    r2=rand;

    I(i)=r2*S(i)/(4*pi*di(i));

end

end

function [X]=initialization(N,dim,up,down)

if size(up,2)==1

    X=rand(N,dim).*(up-down)+down;

end

if size(up,2)>1

for i=1:dim

        high=up(i);low=down(i);

        X(:,i)=rand(N,1).*(high-low)+low;

end

end
```

```
end
```

% hbamlt

```
function[PSNRx,SSIMx]=hbamlt(II,dim)

fitfun =@CEP;

T=100;

Lb=0;

Ub=255;

N=30;

II=imread('1chestXray.jpeg');

[r1,c1,s1]=size(II);

if s1>1

    II=rgb2gray(II);

end

[xmin,fmin,CNVG]=HBA(fitfun,dim,Lb,Ub,T,N,II);

%figure,

%semilogy(CNVG,'r');

%xlim([0 T]);

%title('Convergence curve');

%xlabel('Iteration');

%ylabel('Best fitness obtained so far');

legend('HBA')

%display(['The best location= ', num2str(xmin)]);

%display(['The best fitness score = ', num2str(fmin)]);

TH=sort(round(xmin))

J1=double(II);

[PSNR,J2]=psnrfn(J1,TH);


J2=uint8(J2);

 imshow(J2);
 figure;

% imshow(I);

% figure;

% imhist(J2);

PSNRx=(psnr(II,J2));

SSIMx=(ssim(II,J2));
```

%hbatest2

```
clear all

close all

dim=[3];

II=imread('1chestXray.jpeg');

II=rgb2gray(II);

PS=[];

SS=[];

for ii=1:length(dim)

for kk=1:15

    [PSNRx,SSIMx]=hbamlt(II,dim(ii));

    PS(kk,ii)=PSNRx;

    SS(kk,ii)=SSIMx;

end

end

TH={'Th=2','Th=5','Th=7','Th=9','Th=11'};

anova1(PS,TH)

ylabel('PSNR')

title('1chestXray')

anova1(SS,TH)

ylabel('SSIM')

title('1chestXray')
```

%hbathreshold

```
clc

clear all

close all

objfunc=@CEP;

lb=0;

ub=255;

dim=2;%No. of threshold

N=30;

tmax=100;

[Xprey, Food_Score,CNVG] = HBA(objfunc, dim,lb,ub,tmax,N);
```

%lognew

```
function A=lognew(X,Y)

for i=1:length(X)

if ((X(i)==0)||(Y==0))

        A(i)=0;

else

        A(i)=log(X(i)/Y);

end

end

end
```

%main

```
clc;clear all;

close all;

tic

fitfun =@CEP;
```

```
dim=5;
T=100;
Lb=0;
Ub=255;
N=30;
II=imread('elaine.gif');
[r1,c1,s1]=size(II);
if s1>1
    II=rgb2gray(II);
end
[xmin,fmin,CNVG]=HBA(fitfun,dim,Lb,Ub,T,N,II);
figure,
semilogy(CNVG,'r')
xlim([0 T]);
title('Convergence curve')
xlabel('Iteration');
ylabel('Best fitness obtained so far');
legend('HBA')
display(['The best location= ', num2str(xmin)]);
display(['The best fitness score = ', num2str(fmin)]);
TH=sort(round(xmin))
J1=double(II);
[PSNR,J2]=psnrfn(J1,TH);

J2=uint8(J2);
imshow(J2);
% figure;
% imshow(I);
% figure;
% imhist(J2);
```

```
PSNRx=(psnr(II,J2))

SSIMx=(ssim(II,J2))

toc
```

%psnrfunction

```
function [PSNR,J2]=psnrfn(J1,Th)

Th=[0 Th];

J2=J1;

L=length(Th);

[r,c]=size(J1);

for i=1:L

if i~=L

        f=find((J1>=Th(i))&(J1<Th(i+1)));

        A=J1(f);

        w1(i)=numel(A)/(r*c);

if numel(A)==0

            mu(i)=0;

else

            mu(i)=mean(A);

end

        k=Th(i+1);

        J2(f)=mu(i);

else

        f=find((J1>=Th(i)));

        A=J1(f);

        w1(i)=numel(A)/(r*c);

if numel(A)==0

            mu(i)=0;

else

            mu(i)=mean(A);
```

```
end
        J2(f)=mu(i);
end
end
D=abs(J1(:)-J2(:));
MSE=mean(D.^2);
PSNR=10*log10((255^2)/MSE);
% imshow(uint8(J2))
```

%sumsquare

```
function [y] = sumsqu(xx)
d = length(xx);
sum = 0;
for ii = 1:d
    xi = xx(ii);
    sum = sum + ii*xi^2;
end
y = sum;
end
```

%test1

```
clc
clear all
close all
M=[2,5,7,9,11];
I=imread('1skincancer.jpg');
PS=[];
SS=[];
for ii=1:length(M)
for kk=1:15
    [PSNRx,SSIMx]=ccomlt(I,M(ii));
    PS(kk,ii)=PSNRx;
    SS(kk,ii)=SSIMx;
end
end
TH={'Th=2','Th=5','Th=7','Th=9','Th=11'};
anova1(PS,TH)
ylabel('PSNR')
title('brainMRI')
anova1(SS,TH)
ylabel('SSIM')
title('brainMRI')
```

%testing

```
function [Xprey, Food_Score,CNVG] = HBA(objfunc, dim,lb,ub,tmax,N)
beta       = 6;     % the ability of HB to get the food  Eq.(4)
C       = 2;     %constant in Eq. (3)
vec_flag=[1,-1];
%initialization
X=initialization(N,dim,ub,lb);
%Evaluation
fitness = fun_calcobjfunc(objfunc, X);
[GYbest, gbest] = min(fitness);
Xprey = X(gbest,:);
for t = 1:tmax
    alpha=C*exp(-t/tmax);   %density factor in Eq. (3)
    I=Intensity(N,Xprey,X); %intensity in Eq. (2)
for i=1:N
```

```
        r =rand();

        F=vec_flag(floor(2*rand()+1));

for j=1:1:dim

            di=((Xprey(j)-X(i,j)));

if r<.5

                r3=rand;                        r4=rand;                        r5=rand;


                Xnew(i,j)=Xprey(j) +F*beta*I(i)*
Xprey(j)+F*r3*alpha*(di)*abs(cos(2*pi*r4)*(1-cos(2*pi*r5)));

else

                r7=rand;

                Xnew(i,j)=Xprey(j)+F*r7*alpha*di;

end

end


FU=Xnew(i,:)>ub;FL=Xnew(i,:)<lb;Xnew(i,:)=(Xnew(i,:).*(~(FU+FL)))+ub.*FU+lb
.*FL;


        tempFitness = fun_calcobjfunc(objfunc, Xnew(i,:));

if tempFitness<fitness(i)

            fitness(i)=tempFitness;

            X(i,:)= Xnew(i,:);

end

end

    FU=X>ub;FL=X<lb;X=(X.*(~(FU+FL)))+ub.*FU+lb.*FL;

    [Ybest,index] = min(fitness);

    CNVG(t)=min(Ybest);

if Ybest<GYbest

        GYbest=Ybest;

        Xprey = X(index,:);

end

end
```

```
Food_Score = GYbest;
end


function Y = fun_calcobjfunc(func, X)
N = size(X,1);
for i = 1:N
    Y(i) = func(X(i,:));
end
end
function I=Intensity(N,Xprey,X)
for i=1:N-1
    di(i) =( norm((X(i,:)-Xprey+eps))).^2;
    S(i)=( norm((X(i,:)-X(i+1,:)+eps))).^2;
end
di(N)=( norm((X(N,:)-Xprey+eps))).^2;
S(N)=( norm((X(N,:)-X(1,:)+eps))).^2;
for i=1:N
    r2=rand;
    I(i)=r2*S(i)/(4*pi*di(i));
end
end
function [X]=initialization(N,dim,up,down)
if size(up,2)==1
    X=rand(N,dim).*(up-down)+down;
end
if size(up,2)>1
for i=1:dim
        high=up(i);low=down(i);
        X(:,i)=rand(N,1).*(high-low)+low;
end
```

```
end
end
```

%wave2

```
H=imread('honey badger.png');
imshow(H);
.
```

REFERENCES:

[1] R. Yogamangalam and B. Karthikeyan, “Segmentation techniques comparison in image processing,” *Int. J. Eng. Technol.*, vol. 5, no. 1, pp. 307–313, 2013.

[2] X. H. Han and Y. W. Chen, “Biomedical imaging modality classification using combined visual features and textual terms,” *Int. J. Biomed. Imaging*, vol. 2011, 2011, doi: 10.1155/2011/241396.

[3] M. Eapen and R. Korah, “Medical image segmentation for anatomical knowledge extraction,” *J. Comput. Sci.*, vol. 10, no. 7, pp. 1253–1258, 2014, doi: 10.3844/jcssp.2014.1253.1258.

[4] H. Greenspan and T. Syeda-Mahmood, “Parametric and Non-Parametric Clustering for Segmentation,” in *Biomedical Image Processing*, T. M. Deserno, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 227–250. doi: 10.1007/978-3-642-15816-2_9.

[5] R. Rautray, S. Biswas, R. Dash, and R. Dash, “Fruit Fly Algorithm: A Brief Review,” in *Intelligent and Cloud Computing*, 2021, pp. 525–531.

[6] A. H. Gandomi and A. H. Alavi, “Krill herd: A new bio-inspired optimization algorithm,” *Commun. Nonlinear Sci. Numer. Simul.*, vol. 17, no. 12, pp. 4831–4845, 2012, doi: https://doi.org/10.1016/j.cnsns.2012.05.010.

[7] D. Karaboga and B. Basturk, “Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization,” pp. 789–798.

[8] A. G. Hussien *et al.*, “Crow Search Algorithm : Theory , Recent Advances , and Applications,” vol. 8, 2020, doi: 10.1109/ACCESS.2020.3024108.

[9] A. Mohamad, A. M. Zain, N. E. N. Bazin, and A. Udin, “Cuckoo search algorithm for optimization problems - a literature review,” in *Applied Mechanics and Materials*, 2013, vol. 421, no. April 2014, pp. 502–506. doi: 10.4028/www.scientific.net/AMM.421.502.

[10] S. Arora, J. Acharya, A. Verma, and P. K. Panigrahi, “Multilevel thresholding for image segmentation through a fast statistical recursive algorithm,” *Pattern Recognit. Lett.*, vol. 29, no. 2, pp. 119–125, 2008, doi: 10.1016/j.patrec.2007.09.005.

[11] D. Davendra, I. Zelinka, and G. Onwubolu, “Chaotic optimization,” *21st Eur. Conf. Model. Simul. Simulations United Eur. ECMS 2007*, no. May 2014, pp. 339–344, 2007, doi: 10.7148/2007-0339.

[12] O. Avalos, E. Ayala, F. Wario, and M. Pérez-Cisneros, *An accurate Cluster chaotic optimization approach for digital medical image segmentation*, vol. 33, no. 16. 2021. doi: 10.1007/s00521-021-05771-8.

[13] P. Y. Yin, “Multilevel minimum cross entropy threshold selection based on particle swarm optimization,” *Appl. Math. Comput.*, vol. 184, no. 2, pp. 503–513, Jan. 2007, doi: 10.1016/J.AMC.2006.06.057.

[14] F. A. Hashim, E. H. Houssein, K. Hussain, M. S. Mabrouk, and W. Al-Atabany,

"Honey Badger Algorithm: New metaheuristic algorithm for solving optimization problems," *Math. Comput. Simul.*, vol. 192, pp. 84–110, 2022, doi: 10.1016/j.matcom.2021.08.013.

[15] C. M. Begg, K. S. Begg, J. T. Du Toit, and M. G. L. Mills, "Scent-marking behaviour of the honey badger, Mellivora capensis (Mustelidae), in the southern Kalahari," *Anim. Behav.*, vol. 66, no. 5, pp. 917–929, 2003, doi: https://doi.org/10.1006/anbe.2003.2223.

[16] A. Horé and D. Ziou, "Image Quality Metrics: PSNR vs. SSIM," in *2010 20th International Conference on Pattern Recognition*, 2010, pp. 2366–2369. doi: 10.1109/ICPR.2010.579.

[17] U. Ulusoy, "Application of ANOVA to image analysis results of talc particles produced by different milling," *Powder Technol. - POWDER TECHNOL*, vol. 188, pp. 133–138, 2008, doi: 10.1016/j.powtec.2008.04.036.

[18] https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset?resource=download

[19] https://www.kaggle.com/datasets/paultimothymooney/chest-xray-Pneumonia

[20] https://www.kaggle.com/datasets/wanderdust/skin-lesion-analysis-toward-melanoma-detection

[21] https://www.kaggle.com/paultimothymooney/blood-cells

[22] https://towardsdatascience.com/image-segmentation-part-1-9f3db1ac1c50

[23] https://towardsdatascience.com/image-segmentation-part-2-8959b609d268

[24] https://ieeexplore.ieee.org/document/9381863]

[25] https://doi.org/10.1016/j.advengsoft.2016.01.008

[26] https://ieeexplore.ieee.org/document/7383528

[27] https://petapixel.com/inverse-square-law-light