ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

MSc THESIS

Fatma ÖZYURT

COLOR QUANTIZATION BY PARTICAL SWARM OPTIMIZATION ALGORITHM, DIFFERENTIAL EVOLUTION ALGORITHM AND GREY WOLF ALGORITHM

DEPARTMENT OF COMPUTER ENGINEERING

ADANA, 2017

ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

COLOR QUANTIZATION BY PARTICAL SWARM OPTIMIZATION ALGORITHM, DIFFERENTIAL EVOLUTION ALGORITHM AND GREY WOLF ALGORITHM

Fatma OZYURT

MSc THESIS
DEPARTMENT OF COMPUTER ENGINEERING

We certify that the thesis titled above was reviewed and approved for the award of degree of the Master of Science by the board of jury on 05 / 10 / 2017.

....
Assoc. Prof. Dr. Mustafa ORAL
SUPERVISOR

....
Assist. Prof. Dr. Esra SARAÇ EŞSİZ
MEMBER

....
Assoc. Prof. Dr. Zekeriya TÜFEKCİ
MEMBER

This MSc Thesis is written at the Department of Institute of Natural And Applied Sciences of Çukurova University.
Registration Number:

Prof. Dr. Mustafa GÖK
Director
Institute of Natural and Applied Sciences

ABSTRACT
MASTER THESIS

COLOR QUANTIZATION BY PARTICAL SWARM OPTIMIZATION ALGORITHM, DIFFERENTIAL EVOLUTION ALGORITHM AND GREY WOLF ALGORITHM

Fatma ÖZYURT

ÇUKUROVAUNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES
DEPARTMENT OF COMPUTER ENGINEERING

Supervisor : Assoc. Prof. Dr. Mustafa ORAL
Year: 2017, Pages 99
Jury : Assoc. Prof. Dr. Mustafa ORAL
: Asst. Prof. Dr. Esra SARAÇ EŞSİZ
: Assoc. Prof. Dr. Zekeriya TÜFEKÇİ

Color Quantization is used to reduce the number of colors in an image palette. This process results with shorter codes for each color used in output image in return, it provides compression.

However this process is lossy and strongly effects image quality. As the palette size decreases, image quality detoriates as well.

There are various deterministic algorithms; Uniform Color Quantizing, Median Cut, Octree, etc. for color quantization. Nature inspired algorithms ,however, are not widely studied for quantization purposes. In this thesis, it is aimed to employ some of nature inspired algorithms; Partical Swarm Optimization, Differential Evolution and Grey Wolf Optimization for color quantizing problem.

Implementation details and performance comparisions can be found in the thesis.

Keyword: Color Quantizing, PSO, GWO, DE

ÖZ
YÜKSEK LİSANS TEZİ

PARÇACIK SÜRÜSÜ OPTİMİZASYONU ALGORİTMASI, EVRİMSEL FARK ALGORİTMASI VE BOZKURT ALGORİTMASI İLE RENK KUANTALAMA

Fatma ÖZYURT

ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

Danışman : Doç. Dr. Mustafa ORAL
Yıl: 2017, Sayfa: 99
Jüri : Doç. Dr. Mustafa ORAL
: Yrd. Doç. Dr. Esra SARAÇ EŞSİZ
: Doç. Dr. Zekeriya TÜFEKÇİ

Resmin kalitesinden ödün vermeyerek daha az renk ile gösterilmesi işlemine renk kuantalanması denilmektedir. Resmin daha az renk ile aynı görüntü kalitesi ile gösterilmesi, resmin boyutunu da küçülterek aynı zamanda sıkıştırma yapılmasına da yardımcı olur.

Renk kuantalama işleminde resmin görüntü kalitesi, palet boyutunun küçülmesi ile azalır.

En önemli renk kuantalama algoritmaları; Uniform Renk Kuantalama ,Median Cut ,Octree vb.dir. Literatürde renk kuantalama probleminin çözümü için doğadan esinlenilen algoritmalar çok fazla kullanılmaktadır. Bu tezin amacı doğadan esilenilen bazı meta sezgisel algoritmalar ile renk kuantalama probleminin çözümüdür.

Tezde, kullanılan meta sezgisel algoritmaların renk kuantalanması için yapılan uygulamalarının detayları ve algoritmaların performanslarının sonuçlarına yer verilmiştir.

Anahtar Kelimeler: Renk Kauntalama, PSO, GWO, DE

SUMMARY

Reducing the number of colors of images is called color quantizing. The key factor of color quantizing is selecting the best palette for representing the original image with high quality.

Color quantizing is associated with compression. Reprensenting same picture with less colors also compresses the image.

The aim of thesis is color quantizing by nature inspired algorithms. In this thesis Partical Swarm Optimization, Differencial Evolution Algorithm with Rand_1_Bin, Rand_2_Bin, Best_1_Bin, Best_2_Bin mutation strategies by different crossing over rate values and Grey Wolf Optimization Algorithm is applied to color quantizing process.

In literature color quantizing by nature inspired algorithms had been implemented by post clustering methods.

Most commonly used clustering method is K-Means algorithms. By K-Means algorithm most popular colors and the similiar colors to the pictures popular colors are represented by centroids.

Before color quantizing for initial palettes K-Means algorithms had been iterated several times. By this step post clustering had completed for initial palette.

By initial palette color quantizing process haf been started.In order to get new palettes algorithms steps are applied. Nature inspired algorithms disadvantage is time consuming. Post clustering method also had added ömre time by implemantation of color quantizing.

In this thesis clustering methods had been applied.Only the steps of algorithms are applied. First initial palettes had been created randomly.

As a result of this random initialization step, each algorithms iterations are repeated ten times.For PSO , GWO,DE algorithms repeated up to 100 iteration. Then these iterations repeated ten times 100 iterations.

After these repeated iterations average, standart deviation, median and interquatile ranges are calculated.

100 iterations results and ten times 100 iterations are tested. All results had been closer. The realibility of algorithms had been tested.

Best result had been GWO by 70 wolves, second and third had been PSO and DE respectively.

All results and comparisons are showed in thesis.

ACKNOWLEDGEMENTS

The study presented in this thesis was carried out under the supervision of Associate Prof. Dr. Mustafa Oral. I would like to express my gratitude to him for his supervision guidance, patience, motivation, and his valuable time for this work.

I would also to want to thank valuable my family for their support during this work.

CONTENS **PAGE**

LIST OF TABLES **PAGE**

LIST OF FIGURES

PAGE

LIST SYMBOLS AND OF ABBREVIATIONS

F	**:** Mutation constant
CR	**:** Crossing-Over Rate
DE	**:** Differential Evolution
GWO	**:** Grey Wolf Optimization
PSO	**:** Partical Swarm Optimization
RGB	**:** Red-Green-Blue Dimensions of Colors
RAND_1_BIN	**:** Random to Binary First Mutation Strategy
RAND_2_BIN	**:** Random to Binary Second Mutation Strategy
BEST_1_BIN	**:** Best to Binary First Mutation Strategy
BEST_2_BIN	**:** Best to Binary Second Mutation Strategy
IOR	**:** Interquartile Range
STDDEV	**:** Standart Deviation
MSE	**:** Mean Square Error

1. INTRODUCTION

1.1. What is Color Quantization?

Color quantization is the process of using less colors to demonstrate the same image as high quality as possible to original image. Quantization process associated with the compression of image, the image may differ from the original one while less colors are being used, but the goal is to populate the quality image as similar as possible to the original. The key factor to achiving our goal is the selection of the color palette by choosing the best colors closest to original image.

1.2. Color Spaces and Color Difference

A color space is a point which represent a color in n-dimensional space. For example RGB tuples are a sample of three dimensional color space with three dimensions. These dimesions are red , green , blue. There are many different color spaces. Measurement of color spaces is color difference.The easiest measurement is Euclidan distance for RGB color spaces.

1.3. RGB Color Space

In color spaces RGB (Red-Green-Blue) space is more useful to measure the differences of colors. RGB color space combines red, green, blue dimensions and creates new colors.Hence while creating new color by Red-Green-Blue the range of dimensions must be in (0,255).Colors can not represent out of these range values.In RGB color spaces that color difference is calculated by Euclidian Equation (1).

$$d = \sqrt{(R_1 - R_2)^2 + (G_1 - G_2)^2 + (B_1 - B_2)^2} \tag{1}$$

2. ALGORITHMS

2.1. Partical Swarm Optimization

This algorithm is inspired by Dr. Kennedy and Dr. Eberhart from the behaviors of fishes and birds, which animals lives in a swarm in 1995. [1] It is also one of the meta-heuristic algorithms which inspired from nature.

In Partical Swarm Optimization algorithm random behaviors of the animals in swarm effect the other animals.In swarm each eleman is called partical.The population of this particals is called swarm.

The goal of algorithm is finding partical which has best position in the swarm and forcing the other particals to go through the same way with best partical.In swarm particals' aim is getting much better position by comparing their current position with previos positions and comparing with the partical that has best position in swarm.

The swarm initializes by random beginning positions and velocities. All particals in the swarm new velocities and positions value are determined at the iterations.After this step for each particals local bests (present bests) are selected by comparing the previous velocities.At last step ,best value of local bests is selected as global best in swarm.New velocities and positions is updated by following steps of algorithm:

i. By randomly produced beginning positions and velocities creates the initial swarm.
ii. In swarm evey particals' fitness value is calculated.
iii. For each partical's local best is estimated by comparing previous velocities and positions.
iv. Best local best is selected as global best between recently calculated present bests.
v. Positions and velocities are updated by equation (3).

$$V_{id} = W * V_{id} + c_1 * rand_1 * (P_{id} - X_{id}) + c_2 * rand_2 * (P_{gd} - X_{id})$$
$$X_{id} = X_{id} + V_{id} \tag{3}$$

2.2. Differential Evolution Algorithm

Differential Evolution algorithm is a meta-heuristic and population based algorithm.It had been developed by Storn and Price [2].Although Differential Evolution is a flexible algorithm to solve hard problems successfully.In DE algorithm initial population ,quality of genes are improved by initialization,mutation ,recombination, selection steps during iterations[3].

DE algorithm has three major parametres.First one is the size of population which is represented by NP. Population size gives individual number for the problem. Second important parameter is mutation coefficient which is represented by F. This coefficient effects the creation of new donor vector in algorithm from trial vector and target vector. For efficient solutions from algorithm F coefficient's adviced range is [0.5-1]. [2]

Last major parametre is crossing over rate coefficient (CR).This parametre is used to decision of selecting genes after mutation step for new donor vector.

In DE algorithm initial population is created by random values.According to equation (4) trial vector v_i is created.After creating new trial vector, maximum and minimum range of vector must be checked for every individuals.

$$v_i = x_{r1} + F_1(x_{r2} - x_{r3}) \tag{4}$$

x_{r1} ,x_{r2}, x_{r3} individuals are different from each other and selected by randomly.New individuals of population are created by crossing over rate coefficient from v_i to u_i.

$$\underset{\substack{\uparrow\\ \text{trial}}}{u_{i,j}} = \begin{cases} \overset{\text{donor}}{v_{i,j}} & if \;\; rand_{i,j} \leq \underset{\substack{\uparrow\\ \text{crossover ratio}}}{CR} \;\; or \;\; j = irand \\ \underset{\substack{\uparrow\\ \text{target}}}{x_{i,j}} & else \end{cases} \tag{5}$$

In recombination step of algorithm if random value is less or equal to CR then new individual is selected from donor population (v_i) else it is selected from target population(x_i) for new trial population (u_i).

Last step is selection.In this step donor population's and target population's fitness values are compared. If the fitness value of new population's individual's (u_i) fitness value is less than target population's individual's (x_i) fitness value, the greatest individual is selected for the new population according to equation (6).

$$x_i^{k+1} = \begin{cases} u_i^k & if \;\; f(u_i^k) < f(x_i^k) \\ x_i^k & else \end{cases} \tag{6}$$

2.3. Grey Wolf Algorithm

Grey wolves live in gregariously and it is called pack.Each pack has at least 5 and maximum 12 members. Grey Wolf pack has a stricted hierarchy in order to survive together in pack .There are four level of wolves in pack.

First level wolf is alpha wolf.This wolf decides for hunting, and other needs for pack in order to survive and live together.In hierarchy of the pack ,managemet abilities are more important being the strongest wolf in pack.Alpha wolf is not the strongest wolf in pack.

Second level wolves are beta wolves.They assist alpha wolves to apply whole orders to the pack and give feedback from pack to alpha wolf about the orders.Beta wolves can also give orders other least level wolves.

Third level of wolves are delta wolves. These are scouts, sentinels, elders and hunters. They are allowed to give order to just omega wolves. They are required to listen the orders of alpha and betas. Scouts are responsible for checking the safety of new places to survive. Sentinels are responsible for the safety of current areas that the pack live. Elders decides new alpha and beta wolves. Hunter wolves are responsible for hunting according to direction of alpha wolves.

Last level wolves are omega wolves and they are required to listen the orders of alpha, beta and delta. They are not allowed to refuse the orders.

According to this hierarchy construction in pack wolves are starting hunting by tracking with scouts wolves and they try to get closed to the prey. When they exactly found prey they encircle the prey. Last step in hunting they take the control of prey in order to not escape the prey then wolves could attack the prey. These steps are shown in Figure 2.1[4].

Hunting behaviour of grey wolves: (A) chasing, approaching, and tracking prey (B-D) pursuiting, harassing, and encircling (E) stationary situation and attack

Figure 2.1. Steps of Wolf Attack

In GWO algorithm encircling prey is applied by equation (7) and (8).

$$\vec{D} = |\vec{C}.\overrightarrow{X_p}(t) - \vec{X}(t)| \tag{7}$$

$$\vec{X}(t+1) = \overrightarrow{X_p}(t) - \vec{A}.\vec{D} \tag{8}$$

In formula t represents the iteration, Xp vector is the direction of prey, X vector represent the wolves places $\vec{A}$ and $\vec{C}$ coefficent vector is calculated by equations (9) and (10)

$$\vec{A} = 2\vec{a}.\overrightarrow{r_1} - \vec{a} \tag{9}$$

$$\vec{C} = 2.\overrightarrow{r_2} \tag{10}$$

Vector a is decreased from 2 to 0 by iteration number , r1 and r2 are random values in (0,1) range. Wolves updates their position by equations (7) and (8). During the hunting prey's location is not certain. Hence the alpha wolves position is accepted the best solution for prey location. Because alpha wolf is the closest wolf to the prey.In the hierarchy of pack other wolves are not allowed to attack prey. As a result of this rule whole wolves updates their position according dominant wolves' positions in equation (11).

$$\overrightarrow{D_\alpha} = |\vec{C_1}.\overrightarrow{X_\alpha} - \vec{X}|, \overrightarrow{D_\beta} = |\vec{C_2}.\overrightarrow{X_\beta} - \vec{X}|, \overrightarrow{D_\delta} = |\vec{C_3}.\overrightarrow{X_\delta} - \vec{X}|$$

$$\overrightarrow{X_1} = \overrightarrow{X_\alpha} - \vec{A_1}.(\overrightarrow{D_\alpha}), \overrightarrow{X_2} = \overrightarrow{X_\beta} - \overrightarrow{A_2}.(\overrightarrow{D_\beta}), \overrightarrow{X_3} = \overrightarrow{X_\delta} - \overrightarrow{A_3}.(\overrightarrow{D_\delta})$$

$$\vec{X}(t+1) = \frac{\overrightarrow{X_1} + \overrightarrow{X_2} + \overrightarrow{X_3}}{3} \tag{11}$$

After these calculations when prey's location is found wolves stop changing their positions.Pseudo code of GWO algorithm is reprensented in Figure 2.2

```
Initialize the grey wolf population Xi (i = 1, 2, ..., n)
Initialize a, A, and C
Calculate the fitness of each search agent
Xα=the best search agent
Xβ=the second best search agent
Xδ=the third best search agent
while (t < Max number of iterations)
    for each search agent
            Update the position of the current search agent by equation (3.7)
    end for
    Update a, A, and C
    Calculate the fitness of all search agents
    Update Xα, Xβ, and Xδ
    t=t+1
end while
return Xα
```

Figure 2.2. Pseudo code of GWO algorithm

3. PREVIOUS WORKS

3.1. Color Quantization Based On Partical Swarm Optimization

The main objective of color quantization is mapping the original image by set of colors. The set of color in quantized image had to be smaller than original image [12]. The similarity between the qauntized image and original image must be increased by colormaps [11].

The color quantizing problem is a NP complete problems [8]. Hence in color quantizing it is not possible to reaching the global optimal solution. It needs large amount of time. Enable to solve these types of algorithms different methods are applied. One of the commonly used method is K-means.

K-means method had been used in color quantizing problems [9], [10]. K-means method is started the impelmentation by random initial values. This requirement is leaded to K-means method's results could be less closest the optimal values for color quantizing. Solving this problem stochastic optimization models are applied with K-means in color quantizing.

In this work Partical Swarm Optimization is used with K-means. Partical Swarm Optimization is an algorithm that is inspired by the behaviours of fish and birds to reach the optimal solution. PSO minimizes the effects of initial values to reach optimal solution.

3.1.1. Clustering Approaches

3.1.1.1. Pre-Clustering Approaches

Pre-clustering method is dividing the original image into regions which has different part of image. The colors for colormap can be selected from each region.The division of clustering can be done different methods.

Median Cut Algorithm is used because of easy applications in image processings. It seperates the regions into rectangular boxes by the median of colors up to iteration number.

The varince – based algorithm seperates the regions into rectangular boxes due to mean square errors of boxes between colors.

The octree quantization builds a tree structure containing always a maximum of n different colors. If a further color is to be added to tree structure, its color value has to be merged wtih most likely one that is already added in the tree.The both values are subsituted by their mean.Octree algorithm results are same as Median Cut Algorithm but Octree Algorithm is more faster than Median Cut Algorithm.

The disadvantage of pre-clustering approaches is just working with color spaces of geometric characters.

3.1.1.2. Post-Clustering Approaches

Post-clustering approaches works on clustering by the color spaces. Post – clustering is used by initial random colormaps. Post-clustering methods is forced by iterations to gain much better colormaps. But the most important disadvantage is the time –consuming [11].

The K-means method is the commonly used post-clustering algorithm. K-means strated with initial colormaps. Then each colors in image are assigned into the suitable clusters of K means. The process is iterated until reaching the best results for colormaps. The disadvantage of K means is the dependency of initial colormaps.

Self-Organizing Maps is used for color quantizing. Self-Organizing Maps method changes the initial colormaps by exchancing the colors in image. This process had time consuming problem, hence later the closest colors in image is exchanged with colormaps color.

3.1.2. Color Quantization with Post-Clustering Based on PSO

In color quantizing PSO post-clustering method K-means is applied for new colormaps.In this work fitness function had been the Euclidian Distance of color spaces. At first step of color quantizing centroids of clusters are selected randomly. Then for each pixel the fitness function is calculated.After calculations each pixel is assign to the closest centroid.

At the end of one swarm all centroids are updated by PSO algorithm's velocity and position formulas. Each clusters centroids' local best and global best value is selected in one swarm. At the end of stopping criteria global best cluster is selected as the best colormap in order to quantize image.PSO-CIQ clustering is applied in different K colors.

Here is the psuede code of color quantizing PSO by post-clustering Figure 3.1:

The PSO-CIQ algorithm is summarized below:

1. Initialize each particle by randomly choosing K color triplets from the image.
2. For $t = 1$ to $t_{\max}$
(a) For each particle i
  i. Apply K-means for a few iterations with a probability p_{kmeans}.
  ii. For each pixel z_p
    Calculate $d^2(\boldsymbol{z}_p - \boldsymbol{m}_{i,k})$ for all clusters $\boldsymbol{C}_{i,k}$.
    Assign z_p to $\boldsymbol{C}_{i,kk}$ where
    $$d^2(\boldsymbol{z}_p - \boldsymbol{m}_{i,kk}) = \min_{\forall k=1,\dots,K} \left\{ d^2(\boldsymbol{z}_p - \boldsymbol{m}_{i,k}) \right\}$$
  iii. Calculate the fitness, $f(\boldsymbol{x}_i)$
(b) Find the global best solution $\hat{y}(t)$
(c) Update the centroids using Eq.'s 6 and 7

Figure 3.1. Psuede code of color quantiizng PSO by post-clustering

Here is the fitness function of algorithm. All centroids color spaces is measured according to each pixel.

$$MSE = \frac{\sum_{k=1}^{K} \sum_{\forall z_p \in C_k} (z_p - m_k)^2}{N_p}$$

Figure 3.2. Mean squared error

Table 3.1. Quantization MSE results for the Lenna image using PSO-CIQ

K colors	PSO-CIQ
16	210.203 ± 1.487
32	119.167 ± 0.449
64	77.846 ± 16.132

a) Original Image

b) 16 colors

c) 32 colors

d) 64 colors

Figure 3.3. Quantization results for the Lenna image using PSO-CIQ [2]

3.2. Color Quantization Based On Differential Evolution Optimization

Differential Evolution Algorithm is a population based meta-heuristc algorithm [13-15]. DE algorithm is commonly prefered algorithm because of its simple application, less parameters and fast closed results to local optimal solutions. DE algorithm is a better selection than PSO algorithm enable to solve NP –Hard color quantizing problem [16].

The performance of algortihm is depended on F and CR parametres. Hence it is hard to give decide about the value of these parametres. Because of situation these parametres are decided during the evolution of algorithm. It is called Self–Adaptive Differential Evolution (SaDE-CIQ) [19, 20].

In SaDE-CIQ K-means is applied to for the post clustering of image. K-means algorithm used for the post-clustering to divide the original image into regions. The objective of applying K-means method to performe better selections to create the colormaps.

In SaDE-CIQ post clustering method is used. Hence it is started by random colormaps. After the random initial colormaps a few K-means iterations are applied. By these few iterations first sample colormaps is created and then the implementations of DE is applied . New cluster centroids are decided by the DE operations and parametres.

3.2.1. A Self-Adaptive Mechanic and a Mixed Mechanic of DE

In impelemantation of Differential Evolution it is hard to decide to set the right values of F and CR parametres. Enable to select the best value for these parametres, the solution is calculating the values of F and CR in the implementation of evolutions self-adaptively. In self-adaptively DE F and CR coefficents are calculated by the following equations[19-20]:

$$F^j = \begin{cases} 0.1 + \text{rand} * 0.9, & \text{if rand} < 0.1, \\ F^j, & \text{otherwise,} \end{cases}$$

$$\text{CR}^j = \begin{cases} \text{rand}, & \text{if rand} < 0.1, \\ \text{CR}^j, & \text{otherwise.} \end{cases}$$

Figure 3. 4.

The disadvantage of selecting the values of parametres self-adaptively could effect negatively performance of DE. Enable to decrease the effect of disadvantage K-means algorithm is applied. K-means is applied to select the most suitable colormaps for color quantizing by DE. K-means Algorithm is improved the selection of optimal colormaps to quantize the original image.

3.2.2. Color Quantizing Based on A Self-Adaptive DE

In the implemention of selection of optimal colormaps to quantize the original image. By meta-heuristic algorithms time–consuming is the most common problem. According to solve this problem post clustering and pre clustering methods are applied. These clustering methods are performed by the color spaces of colors [21]. In meta–heurisctic algorithm DE applications, in order to solve time-consuming problem post-clustering methods are applied.

In NP-Hard color quantizing problem, meta-heuristic. Differential Evolution algorithm is applied. In order to solve time consuming problem post clustering method is applied by K-means algorithm.

In post-clustering by K-means initial colormaps are used by K colors. Each color in original image is assigned the most suitable clusters according to their color distance.

Color spaces is calculated by the the fitness function. The euclidan equation is used as the fitness function in order to calculate the color spaces of colors. By K-means algorithm original image is divided into K clusters, then clusters are improved by the equations of DE algorithm. Here is the pseude code of color quantizing of Self-Adaptive DE (SaDE-CIQ) algorithm Figure 3.5

Input color image $I = \{z\}$
Set parameters $NP, t_{\max}, p$
$x_i^{j,0} = \mathrm{rand}(0,1) \cdot 255, i = 1, 2, \ldots, D$
$x^{j,0} = \left(x_1^{j,0}, x_2^{j,0}, \ldots, x_D^{j,0}\right), j = 1, 2, \ldots, NP$ //population initialization
$F = 0.5, \mathrm{CR} = 0.6$
for $t = 0$ to $t_{\max}$
  Select a small number of individuals from current population according to p, and adjust them by K-means.
  for $j = 1, 2, \ldots, NP$
    $rnbr_j = \mathrm{rand}(1, D)$
    if $x^{r_2,t} - x^{r_3,t} \leq 100$
      $u^{j,t} = x^{r_1,t} + F \cdot (x^{r_2,t} - x^{r_3,t})$
    else $u^{j,t} = x^{r_1,t} + F \cdot [(x^{r_2,t} - x^{r_3,t}) \% 100]$
    Update F by formulas (5)
    for $i = 1, 2, \ldots, D$
      if $u_i^{j,t} < 0$ then $y_i^{j,t} = 0$
      else if $u_i^{j,t} > 255$ then $y_i^{j,t} = 255$
        else $y_i^{j,t} = u_i^{j,t}$
        end if
      end if // mutation
      $\mathrm{rand}_i = \mathrm{rand}(0,1)$
      if $\mathrm{rand}_i \leq \mathrm{CR}$ or $i = rnbr_j$ then $z_i^{j,t} = y_i^{j,t}$
      else $z_i^{j,t} = x_i^{j,t}$ //crossover
    $z^{j,t} = \left(z_1^{j,t}, z_2^{j,t}, \ldots, z_D^{j,t}\right)$
    Update CR by formulas (6)
    Calculate $g\left(x^{j,t}\right)$ and $g\left(z^{j,t}\right)$
    if $g\left(x^{j,t}\right) > g\left(z^{j,t}\right)$ then $x^{j,t+1} = z^{j,t}$
    else $x^{j,t+1} = x^{j,t}$ //selection
Find the global optimal solution $x^{\text{best}} = \left(x_1^{\text{best}}, x_2^{\text{best}}, \ldots, x_D^{\text{best}}\right)$
Output the optimal colormap $\{c_1, c_2, \ldots, c_K\}, c_k = \left(x_{1+3(k-1)}^{\text{best}}, x_{2+3(k-1)}^{\text{best}}, x_{3+3(k-1)}^{\text{best}}\right), k = 1, 2, \ldots, K$
Construct the quantized image I'

Figure 3.5. Pseude code of CQ Self-Adaptive DE (SaDE-CIQ) algorithm

a) MSE=17.46

b) MSE=22.74

c) MSE= 12.97

d) MSE=8.24

Figure 3.6. Quantization results for SaDE-CIQ [2]

4. MATERIAL AND METHOD

4.1. Implementation of Partical Swarm Optimization

In PSO algorithm the swarm has members that is called particals. The aim of the algorithm is calculating the fastest and best partical according to swarm. Other particals velocities and positions are calculated according to the best partical.In calculations of best particle, each particals' present best velocities are compared with their previous velocities. Particals are updated by best velocity as their new velocity. Then the positions of particals are calculated and updated according to best particle.

In color quantizing by PSO, each partical is accepted a palette. In this research swarm is occurred by color palettes as its particals. In the survey each palette had eight colors. Steps of swarm velocity calculations are applied to achive new best palette.

PSO is a meta-heuristic algorithm.[1] Hence the initialization step of swarm is applied by creating new random colors for new palettes. Each color had three Red-Green-Blue components in (0,255) range randomly. Then new colormaps are created by their new colours.

In color quantizing of PSO fitness function had been used as the color difference of euclidian equation function. The color difference is measured between two colors in fitness function. One of these colors is indicated each palettes of color and also the closest color of palette is shown as the second color. Difference of each color is summed up then the result gave the error of palette according to the image.Then the error calculations of each palette is divided into 1000 in order to show in small number range enable to represent in graphics.

The initialization step of algorithm is started by given a random constant number for velocity. First randomly created palettes had also been the local bests of swarm. In order to calculate the global best of palettes fitness function is applied each palette. Least error palette is chosen as the global best of swarm. Each iteration is updated by new best of locals and globals.

After the initialization step iteration had started in order to calculate new velocitis and positions of particals by equation (3).In color quantizing by PSO after eight colors' RGB values are set by new ones with equation (3) one partical's velocity calculation is completed.

In this step partical's present velocity and previous velocity are compared by fitness function. In this work calculation of partical's velocity means creating new palette. Hence in fitness function new palette and previous palette error is measured. Least error palette is chosen as local best of particals. At the end of iteration global best is selected from best of locals.

In color quantizing by PSO global best of partical means best palette in order to quantize the image.In this survey PSO algorithm is implemented by 10 and 20 particals.

4.2. Implementation of Differential Evolution

One of another meta-heuristic algorithm is Differential Evolution Algorithm. It has three major parametres as NP population size, F mutation coefficient and CR crossing over-rate coefficient.In color quantizing processes the effects of these parametres which are calculated in different mutation strategies.

In color quantizing by DE works NP: 30, F: 2 and CR coefficient in (0, 1) range .CR coefficient had been changed in (0, 1) range in different mutation strategies of DE.

In this work every individuals of population is accepted as palette.Each palette had eight colors. These colors of palette is accepted a gene of individuals. Each of them had eight genes. As a result of these acception for all steps of DE about creating new population is applied each colors in order to get new palettes. After creating eight colors a palette had created as the individuals of initial target population.

In color quantizing DE algorithm fitness function is the color difference by euclidian equation. Color difference is calculated by differences of Red-Green-Blue component of colors by Euclidian Equation (1). At last step of palette error calculation,results are divided into 1000 to represent in graphic into small range numbers.In mutation step each color's RGB components are calculated by the mutation equation formulas in order to estimate new colors as new genes in algorithm.

In mutation step of DE random palettes are selected in order to mutate the gene and create a donor gene.

In this step of algorithm, different mutation strategies are applied to the individuals in order to create new donor population. These strategies are RAND_1_BIN, RAND_2_BIN, BEST_1_BIN, BEST_2_BIN.The performance of these mutation strategies are measured by different CR values in (0,1) range.

In recombination step there was a decision by a random value according to equation (5). If random value less or equal to CR,new individual is selected from donor gene else new gene is selected from target gene for new donor gene. In color quantizing of DE each color of palette is calculated by the formulas of mutation strategies. Then new palette is created. In recombination step if CR less or equal to random value new gene is chosen from donor population else random gene from trial population is choosen. By repeating this step for eight times an individual (a palette) is created.

In the last step of selection, new created palettes error is calculated according to the image by fitness function. Least error palette is selected as the best individual of population.

After all steps of application of DE the image is redrawn by best palette and quantized by less colors.

DE is a meta-heuristic algorithm.[2].This means initial values of algorithm is selected randomly.In order to test the reliability of algorithm for every iteration in every CR values are repeated ten times more.

4.2.1. Rand_1_Bin Mutation Strategy

In color quantizing DE after initialization steps Rand_1_Bin mutation step is applied by formula (12):

$$v_i = x_{r1} + F_1(x_{r2} - x_{r3}) \tag{12}$$

New individuals are selected. According to the formula with randomly selected individuals and then new individuals are created.

In color quantizing DE Rand_1_Bin mutation strategies performance is measured by changing CR coefficient value in (0, 1) range.

4.2.2. Rand_2_Bin Mutation Strategy

In color quantizing DE after the selection of random individuals (palettes).New individuals are created by equation (13) :

$$v_i = x_{r1} + F_1(x_{r2} - x_{r3} + x_{r4} - x_{r5}) \tag{13}$$

4.2.3. Best_1_Bin Mutation Strategy

In color quantizing DE after the selection of random individuals (palettes). New individuals are created by equation (14) :

$$v_i = x_{best} + F_1(x_{r2} - x_{r3}) \tag{14}$$

$X_{\mathbf{best}}$ value is selected by least error palette in current iteration.In every iteration is also updated by better palette.

4.2.4. Best_2_Bin Mutation Strategy

In color quantizing DE after the selection of random individuals (palettes).New individuals are created by equation (15) :

$$v_i = x_{best} + F_1(x_{r2} - x_{r3} + x_{r4} - x_{r5}) \tag{15}$$

X_{best} value is selected by least error palette in current iteration.In every iteration is also updated by better palette.

4.3. Color Quantizing By Grey Wolf Optimization Algorithm

In this task color quantizing procces is applied by the hunting strategy of grey wolves. Achiving best palette for color quantizing, color differences are calculated by RGB color space. By euclidian equations whole pixels' color difference is measured by Red-Green-Blue components of each color.

Similar colors between palettes and the image had low differences. Each palette differences between the image is measured by colors. In GWO color quantizing euclidian function had been fitness function.

By fitness function in GWO, each palatte's error is calculated by the color differences between palettes.

In implemetation of GWO to the color quantizing, each wolf is accepted as a palette. GWO algorithm aim is to find the alpha wolf position which is the closest wolf to the prey after all iterations. Alpha wolf position is accepted the prey's location [1]. As the result of this rule of algorithm alpha wolf position is the best palette for color quantizing GWO. The goal of this survey is searching the best palette like wolf's prey searching.

According to priority rule of wolves dominant palettes are also named alpha, beta and delta palettes respectively.

During the calculations of best palette every steps searching prey steps of GWO is applied to the color quantizing, creating new colors in order to gain new palettes.

As mentioned before each palette is accepted as wolf. In GWO while wolves are searcing a prey and they could search X and Y coordinate to find the prey. In color quantizing GWO that each palette had eight color which means each of wolves had eight dimensions. In order to find more qualified and clear results palettes had eight dimensions.

In initialization of palettes (wolves), colors are selected by randomly in (0,255) range. Each palette had eight colors and each color had Red-Green-Blue three dimensions.

Each dimension of colors are selected randomly then new color is created. During color creating, maximum range had been less and equal to 255 and minimum range greater and equal to 0. Otherwise no color can be represented out of those ranges. After eight colors are finished then new palette had been created.

After initialization of palettes and each error of palettes are calculated according to the original image. At last step of palette error calculation,results are divided into 1000 to represent in graphic into small range numbers.

Before the iterations least error palette had selected as alpha wolf , second least error palette is selected as beta wolf and third least error palette is selected as delta wolf. During the iterations colors of palettes (wolf's position) are updated by the equation (7) and (8).

In each iterations each palette is created by the steps of GWO . (Equation (7), (8), (9), (10), (11)). New palette's error is calculated by fitness function and sorted by their error value. First three palettes which have least errors , are selected new alpha palette ,beta palette, delta palette respectively. At the beginning of new iterations whole dominant wolves'positions are updated then other wolves are updated their location by new dominant position of wolves.

At the end of iteration number best alpha wolf is selected as the best palette that had least error palette between original images in order to quantize the image.

5. RESEARCH AND DISCUSSION

5.1. Results of Color Quantizing Based on PSO

PSO algorithm is also a meta-heuristic algorithm. [1] This means the initialization step of algorithm is implemented by random values. Hence enable to test the realiability of algorithm each niteration number is repeated ten times. The algorithm is tested to work properly via this experiment.

The n times iterations' results and n*ten iterations had gave the similar results. It is proofed that in color quantizing PSO when iteration and partical numbers is increased and error of global best is decreased.

The least palette error is achived by PSO in color quantizing .The main disadvantage of PSO have been time consumption. Because of implementation duration, wents on long hours, only 20 particals can be computed in the algorithm.

At last step of palette error calculation, results are divided into 1000 to represent in graphic into small range numbers. Figure 5.1 represents the n times iterations of PSO :

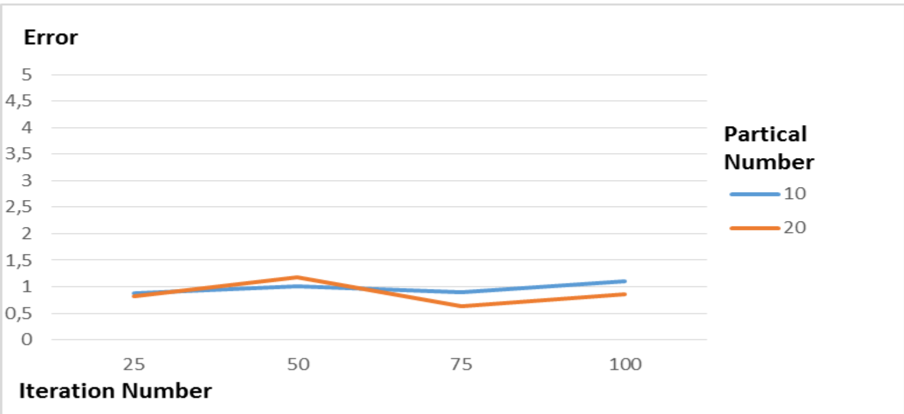

Figure 5.1. Iterations of n times PSO (Flowers)

Figure 5.2 Average of n *ten times iterations of PSO (Flowers) :

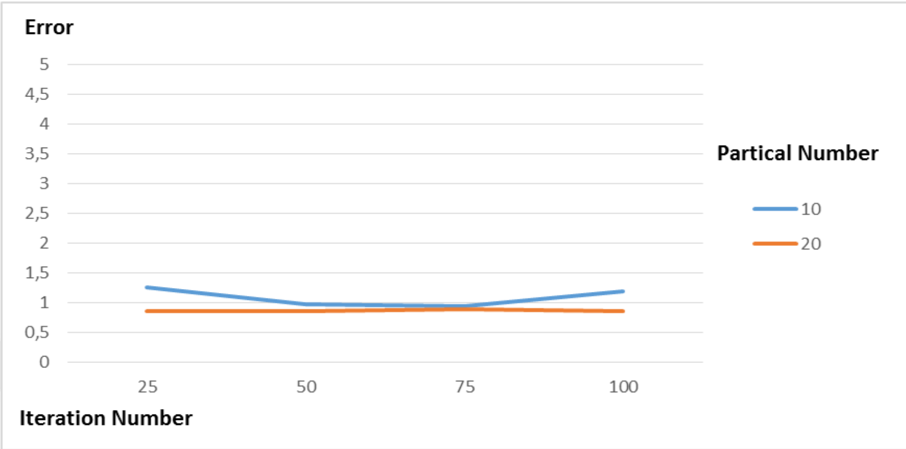

Figure 5.2. Average of n*ten times iterations of PSO (Flowers)

Figure 5.3 represents Median of n*ten times iterations of PSO (Flowers) :

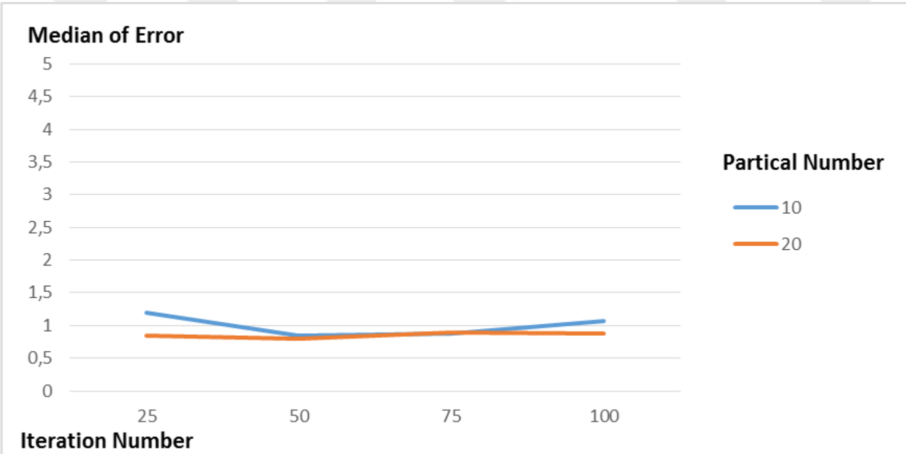

Figure 5.3. Median of n*ten times iterations of PSO (Flowers)

Figure 5.4 represents STDDEV of n*ten times iterations of PSO (Flowers):

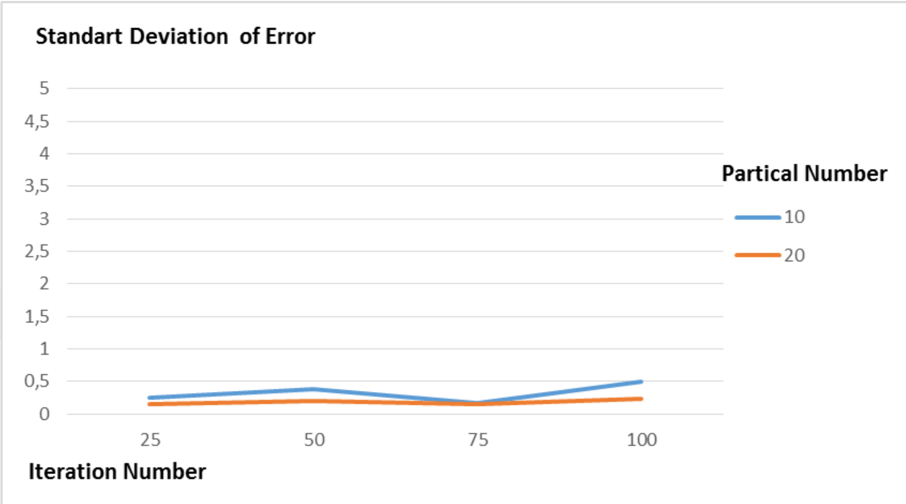

Figure 5.4. STDDEV of n*ten times iterations of PSO (Flowers)

Figure 5.5 represents IOR of n*ten times iterations of PSO (Flowers):

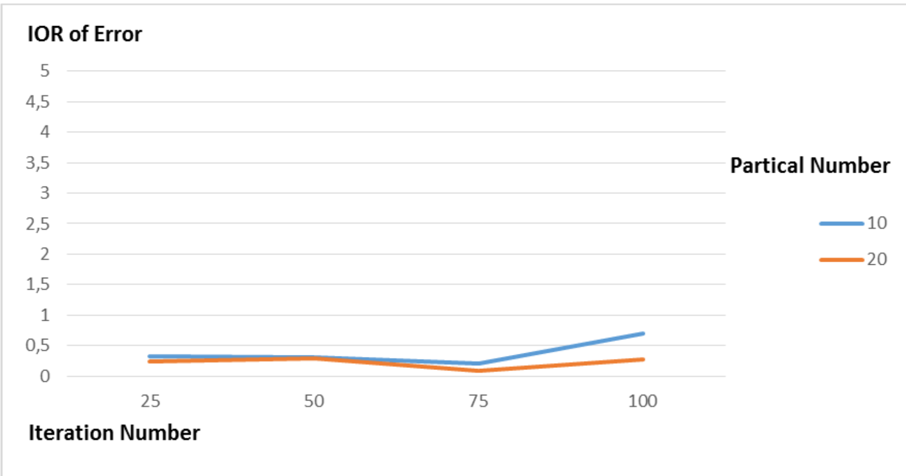

Figure 5.5. IOR of n*ten times iterations of PSO (Flowers)

Figure 5.6 represents Original Image Flowers

Figure 5.6. Original Image Flowers

Figure 5.7 represents PSO quantized image at 100 iteration at 20 particals

Figure 5.7. PSO quantized image at 100 iteration at 20 particals

Figure 5.8 represents Iterations of n times PSO (Lenna):

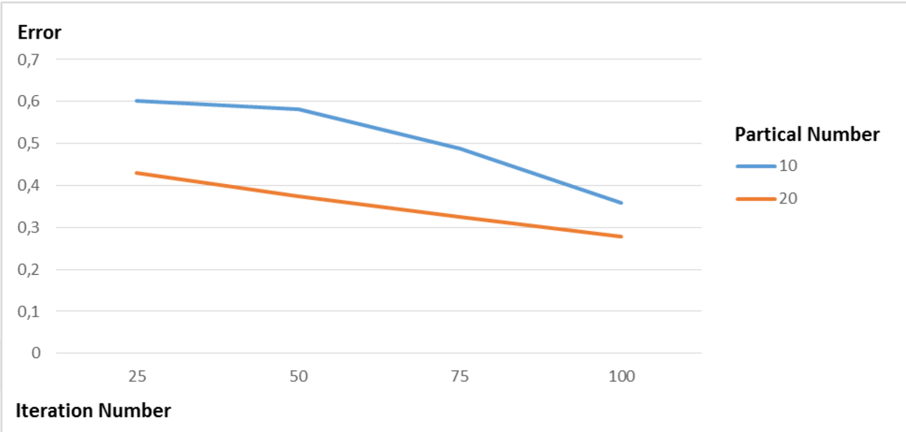

Figure 5.8**.** Iterations of n times PSO (Lenna)

Figure 5.9 represents the average of n*ten times iterations of PSO (Lenna):

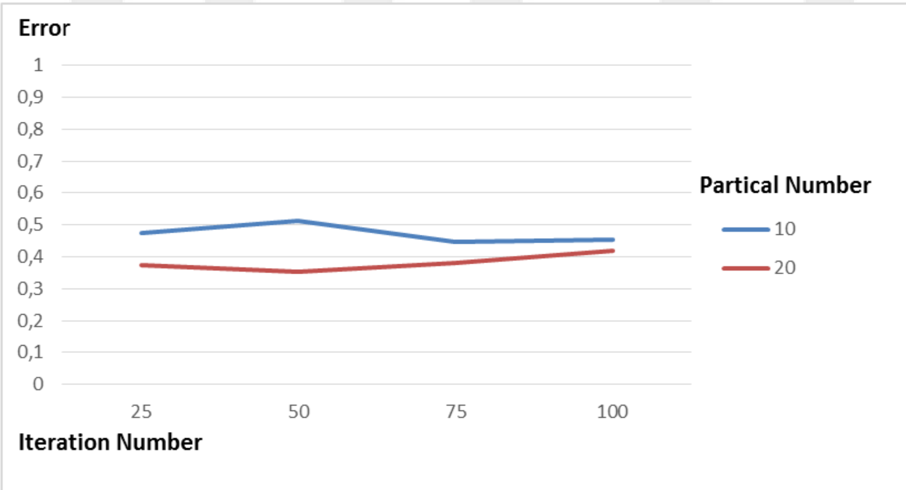

Figure 5.9. Average of n*ten times iterations of PSO (Lenna)

Figure 5.10 represents the median of n*ten times iterations of PSO (Lenna):

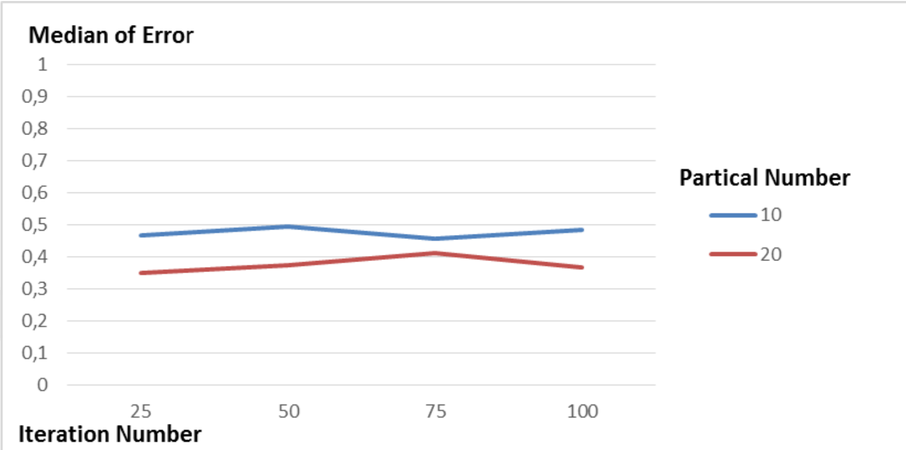

Figure 5. 10. Median of n*ten times iterations of PSO (Lenna)

Figure 5.11 represents the STDDEV of n*ten times iterations of PSO (Lenna):

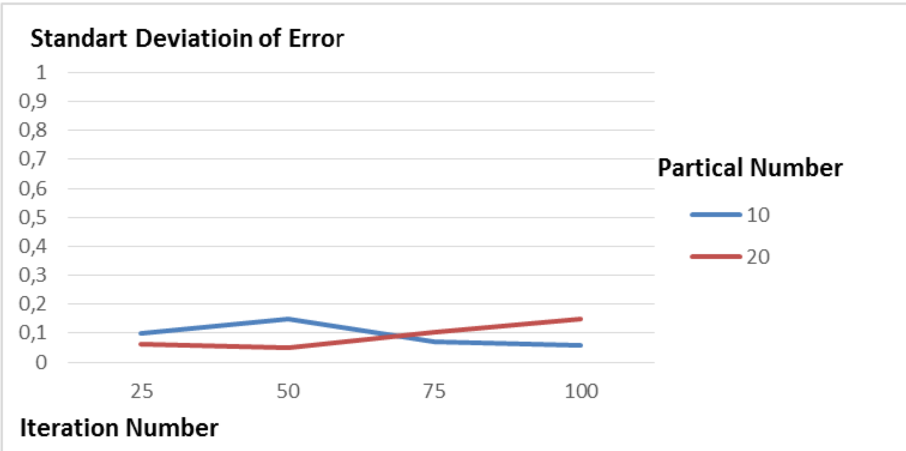

Figure 5. 11. STDDEV of n*ten times iterations of PSO (Lenna)

Figure 5.12 represents the IOR of n*ten times iterations of PSO (Lenna):

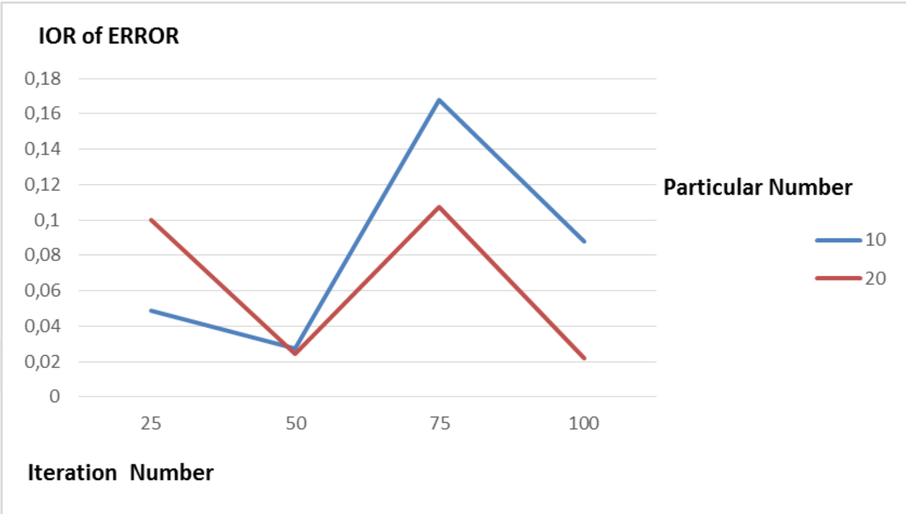

Figure 5.12**.** IOR of n*ten times iterations of PSO (Lenna)

Figure 5.13 represents Original Image Lenna

Figure 5.13. Original Image Lenna

Figure 5.14 represents PSO quantized image at 100 iteration at 20 particals (Lenna)

Figure 5. 14. PSO quantized image at 100 iteration at 20 particals (Lenna)

Figure 5.15 represents Iterations of n times PSO (Baboon):

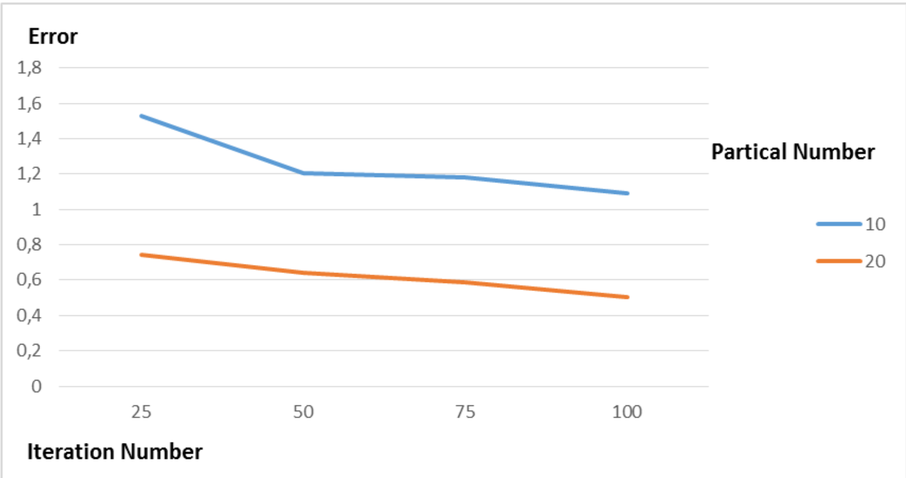

Figure 5.15. Iterations of n times PSO (Baboon)

Figure 5.16 represents the average of n times iterations of PSO (Baboon):

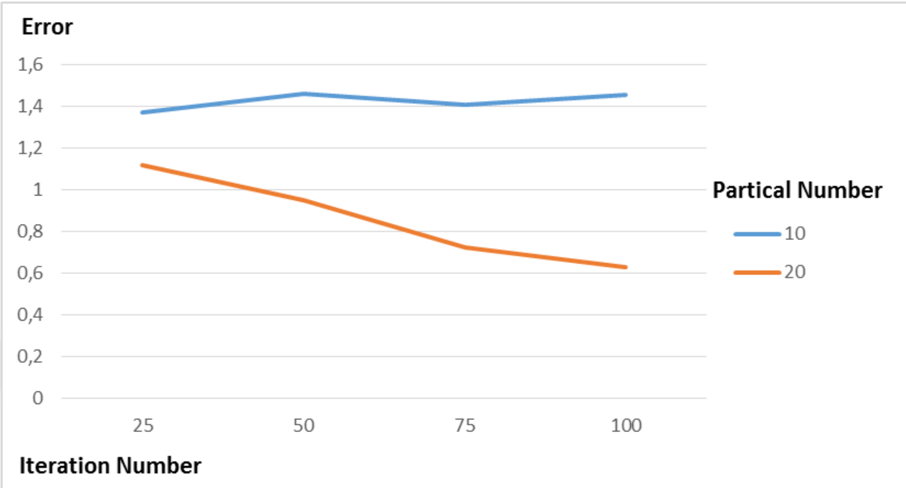

Figure 5.16. Average of n*ten times iterations of PSO (Baboon)

Figure 5.17 represents the median of n*ten times iterations of PSO (Baboon):

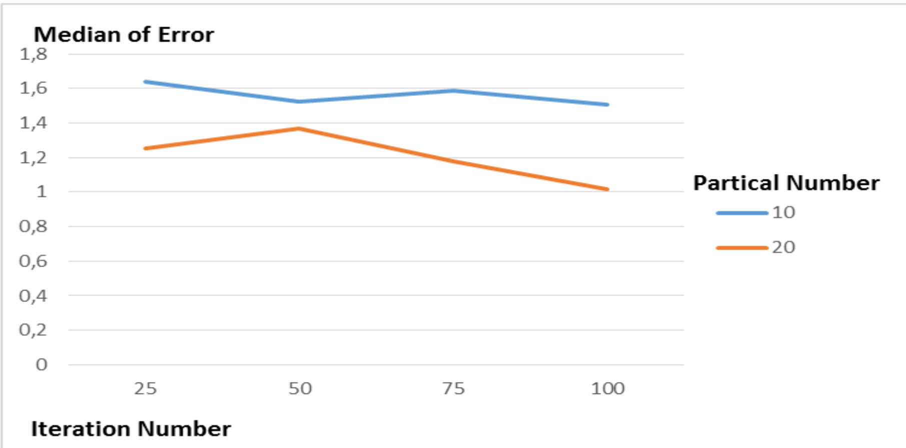

Figure 5.17. Median of n*ten times iterations of PSO (Baboon)

Figure 5.18 represents the STDDEV of n*ten times iterations of PSO (Baboon):

Figure 5. 18. STDDEV of n*ten times iterations of PSO (Baboon)

Figure 5.19 represents the IOR of n*ten times iterations of PSO (Baboon):

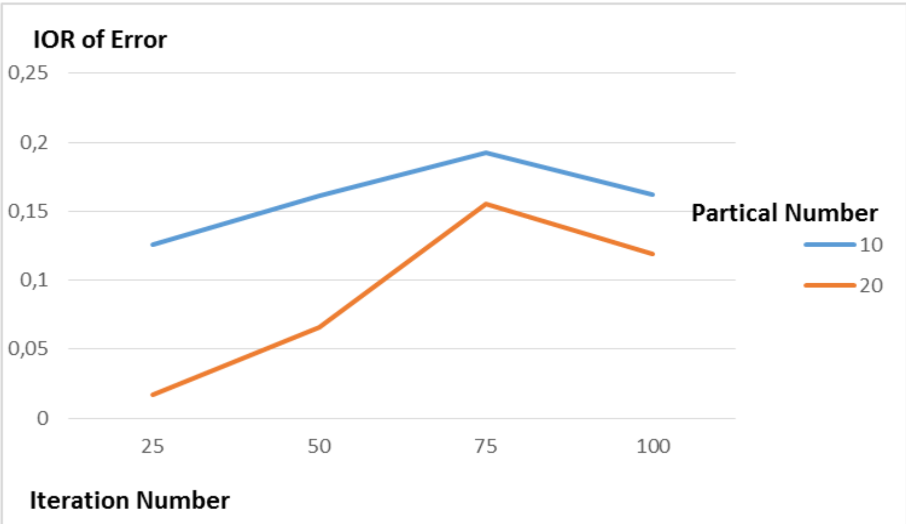

Figure 5.19. IOR of n*ten times iterations of PSO (Baboon)

Figure 5.20 represents Original Image Baboon

Figure 5.20. Original Image Baboon

Figure 5.21 represents PSO quantized image at 100 iteration at 20 particals

Figure 5.21**.** PSO quantized image at 100 iteration at 20 particals

5.2. Results of Color Quantizing Based on DE

DE is meta-heuristic algorithm. Hence the implementation of algorithm had been started by initial random values. Because of these situation in order to test the reliability of experiment every iteration at every CR values are iterated number * ten times.

5.2.1. Rand_1_Bin Mutation Strategy

Rand_1_Bin mutation strategy number of iterations had repeated ten times then the average , median , standard deviation , interquartile range of these values are calculated.In the graphics error had been divided into 1000 in order to show error in small ranges.

Researches showed that in Rand_1_Bin mutation strategy at CR: 0.2 in 50th iteration at Flowers, CR: 0.1 at 25th iteration at Lenna, CR: 0.3 at 50th iteration at Baboon. Averge, median, standart deviation and interquartile range shows the details in iterations. By average values, it is seen that best values in Rand_1_Bin mutation strategy had reached at CR: 0.2 and 0.8 values at 75 th iterations for Flowers image, CR: 0.2 at 100th iterations for Baboon image and CR:0.2 at 100th iterations for Lenna image.

Figure 5.22 shows n times iterations' values of Rand_1_Bin strategy at flowers picture:

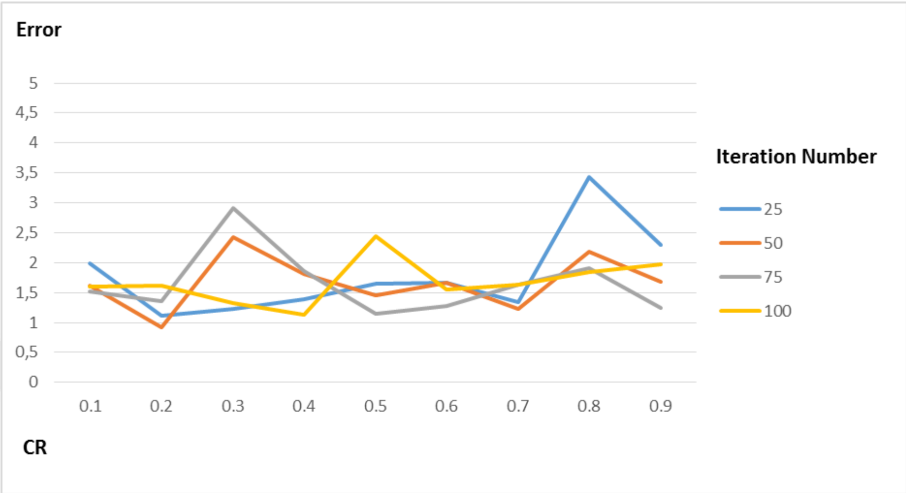

Figure 5. 22**.** n times iterations' results of Rand_1_Bin strategy (Flowers)

Figure 5.23 represents the average of n iterations' results of Rand_1_Bin (Flowers):

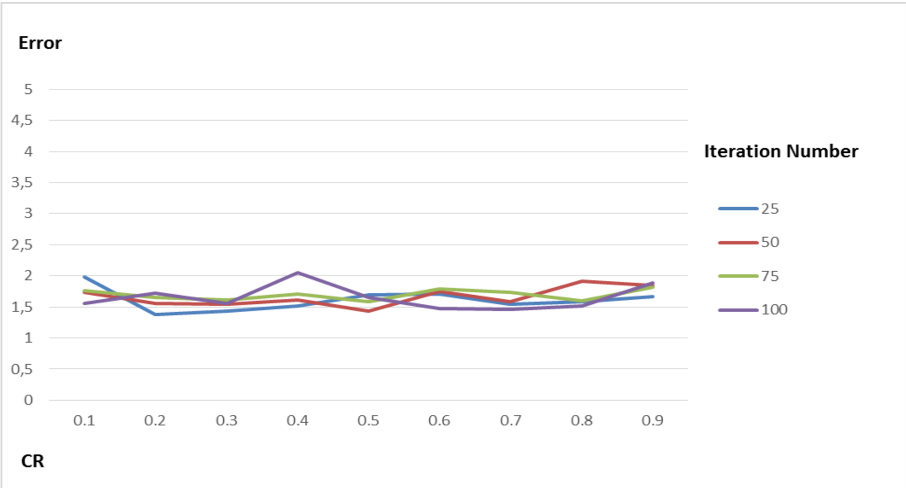

Figure 5.23. Average of n iterations results of Rand_1_Bin (Flowers)

Figure 5.24 represents the median of n*ten times iterations' results of Rand_1_Bin (Flowers):

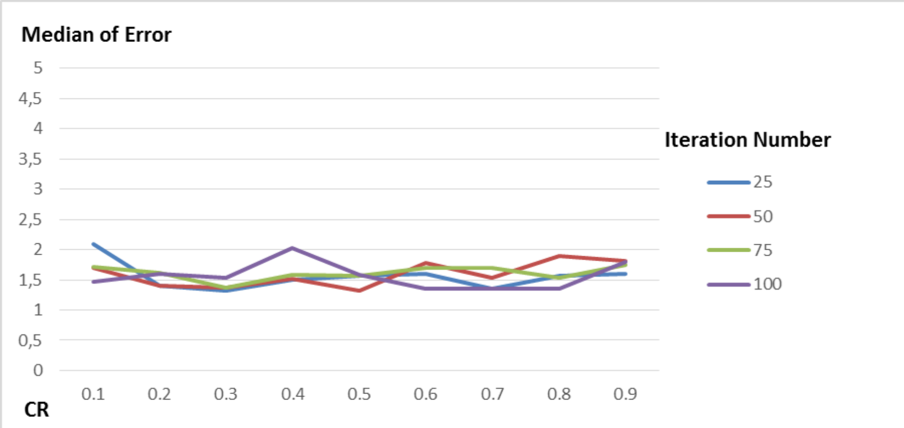

Figure 5.24. Median of n *ten times iterations' results of Rand_1_Bin (Flowers)

Figure 5.25 represents the STDDEV of n *ten times iterations' results of Rand_1_Bin (Flowers):

Figure 5.25. STDDEV of n *ten times results of Rand_1_Bin (Flowers)

Figure 5.26 represents the IOR of n*ten times iterations' results of Rand_1_Bin (Flowers):

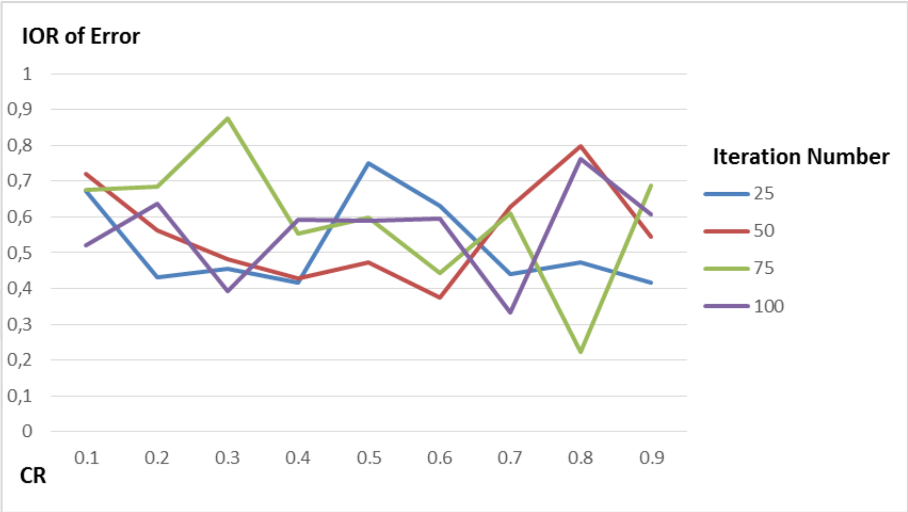

Figure 5.26. IOR of n *ten times iteration results of Rand_1_Bin (Flowers)

Figure 5.27 represents Original Image Flowers

Figure 5.27. Original Image Flowers

Figure 5.28 represents Rand_1_Bin quantized image at CR: 0.2 50th iteration

Figure 5.28**.** Rand_1_Bin quantized image at CR: 0.2 50th iteration

Figure 5.29 represents n times iterations' values of Rand_1_Bin strategy (Lenna):

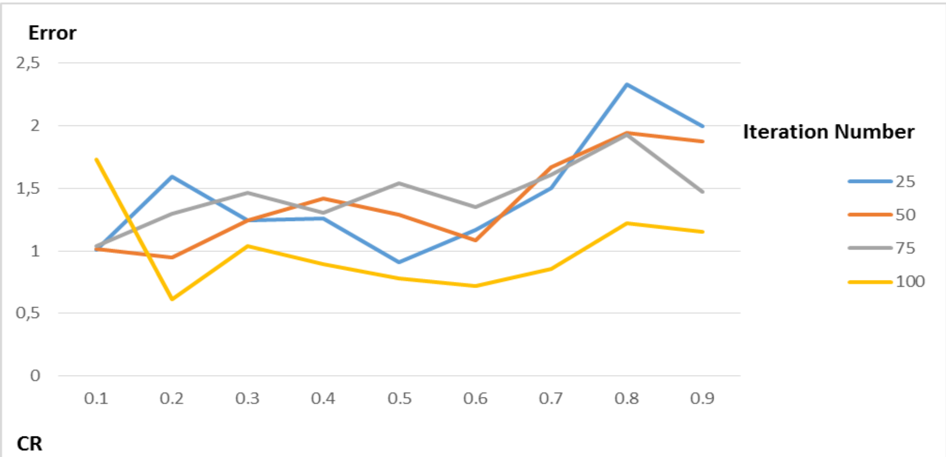

Figure 5.29. n times iterations' values of Rand_1_Bin strategy (Lenna):

Figure 5.30 represents the average of n iterations' results of Rand_1_Bin (Lenna)

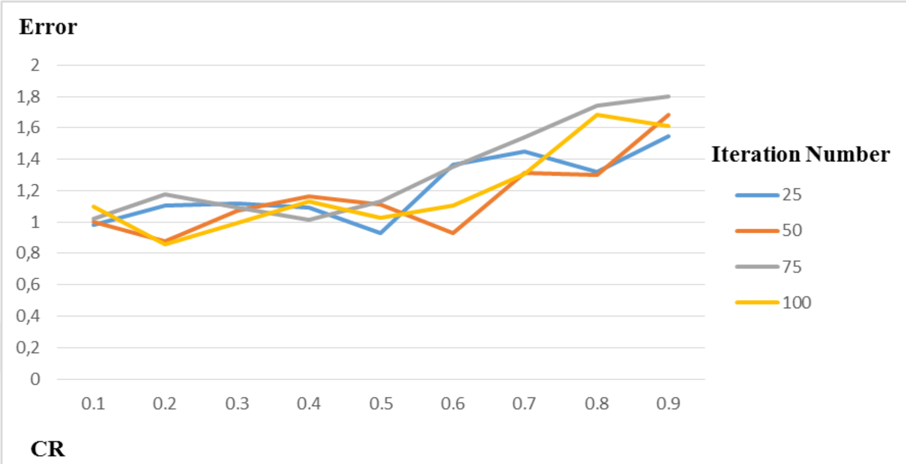

Figure 5. 30. Average of n iterations results of Rand_1_Bin (Lenna)

Figure 5.31 represents the median of n*ten times iterations' results of Rand_1_Bin (Lenna):

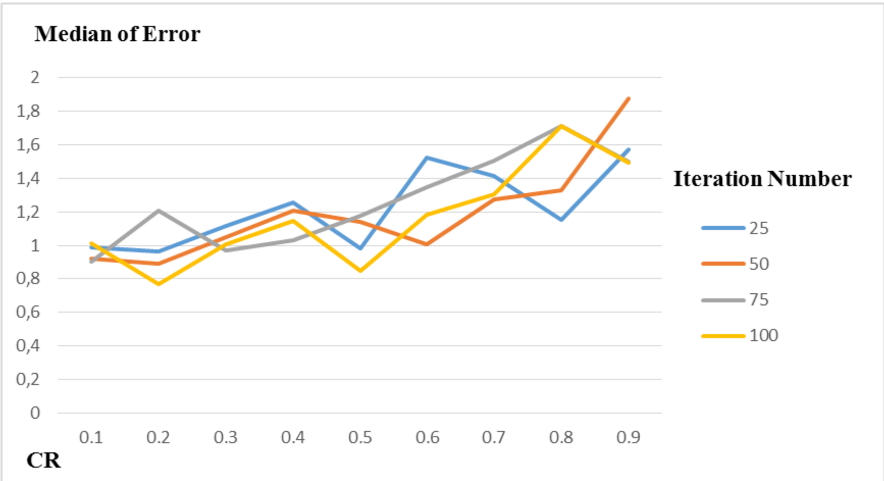

Figure 5.31. Median of n *ten times iterations' results of Rand_1_Bin (Lenna)

Figure 5.32 represents the STDDEV of n *ten times iterations' results of Rand_1_Bin (Lenna):

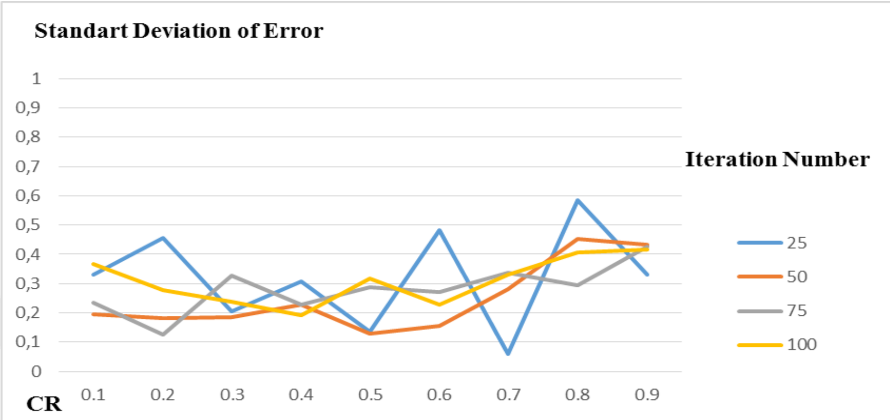

Figure 5.32. STDDEV of n *ten times results of Rand_1_Bin (Lenna)

Figure 5.33 represents the IOR of n*ten times iterations' results of Rand_1_Bin (Lenna)

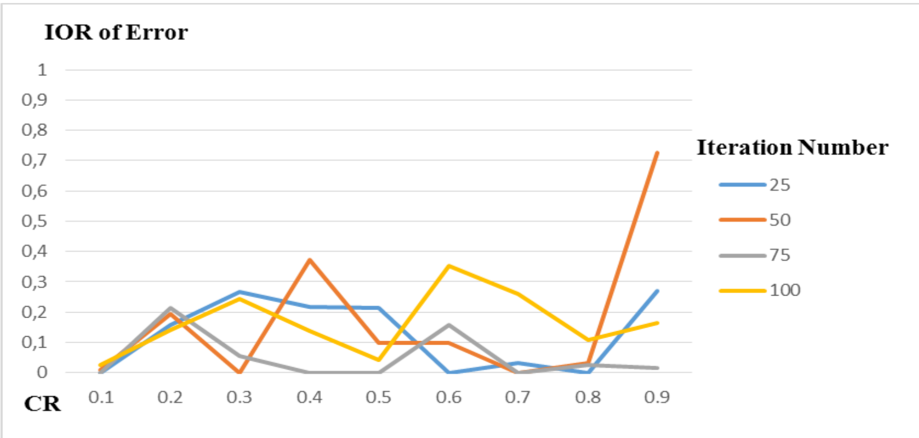

Figure 5.33. IOR of n *ten times iteration results of Rand_1_Bin (Lenna)

Figure 5.34 represents Original Image Lenna

Figure 5.34. Original Image Lenna

Figure 5.35 represents Rand_1_Bin quantized image at CR: 0.1 100th iteration

Figure 5.35. Rand_1_Bin quantized image at CR: 0.1 100th iteration

Figure 5.36 represents n times iterations' values of Rand_1_Bin strategy (Baboon):

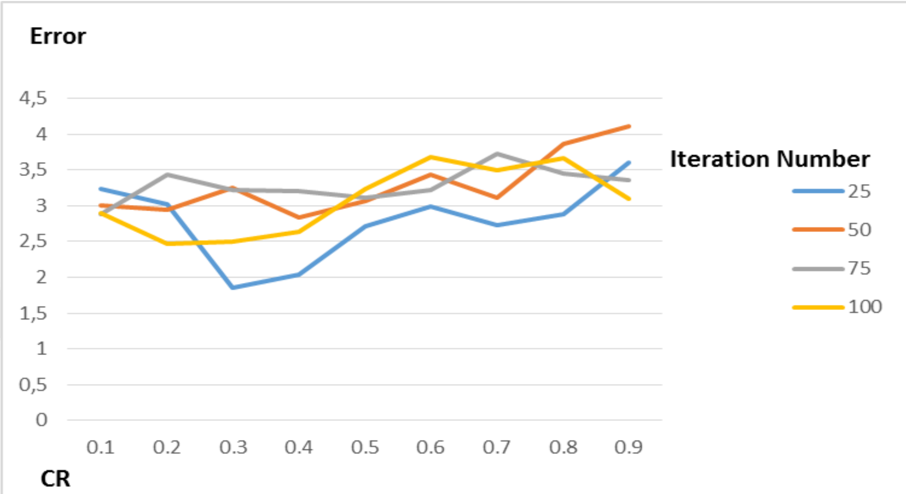

Figure 5. 36. n times iterations' values of Rand_1_Bin strategy (Baboon)

Figure 5.37 represents the average of n iterations' results of Rand_1_Bin (Baboon)

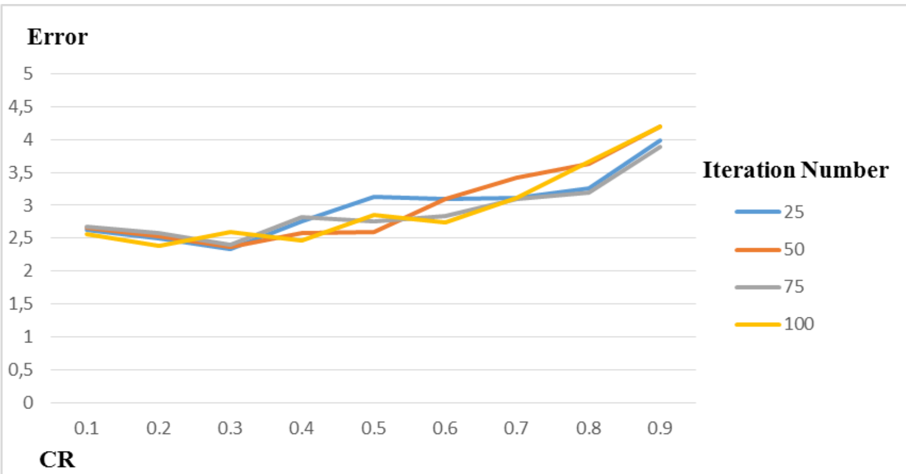

Figure 5.37. Average of n iterations results of Rand_1_Bin (Baboon)

Figure 5.38 represents the median of n*ten times iterations' results of Rand_1_Bin (Baboon):

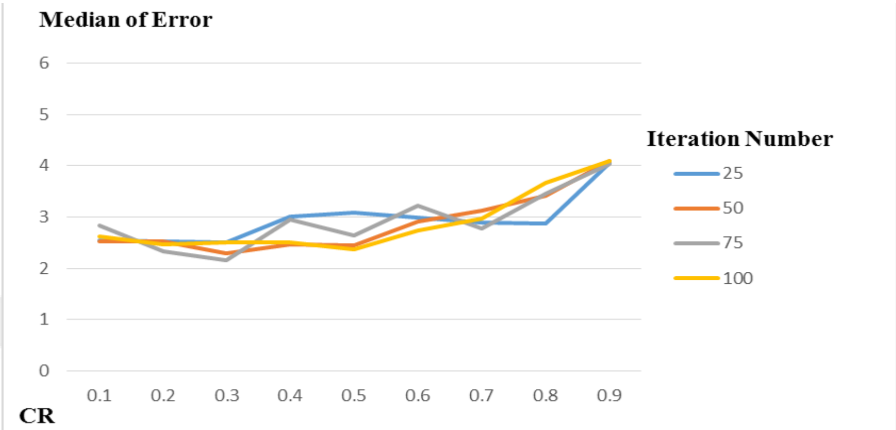

Figure 5.38. Median of n *ten times iterations' results of Rand_1_Bin (Baboon)

Figure 5.39 represents the STDDEV of n *ten times iterations' results of Rand_1_Bin (Baboon):

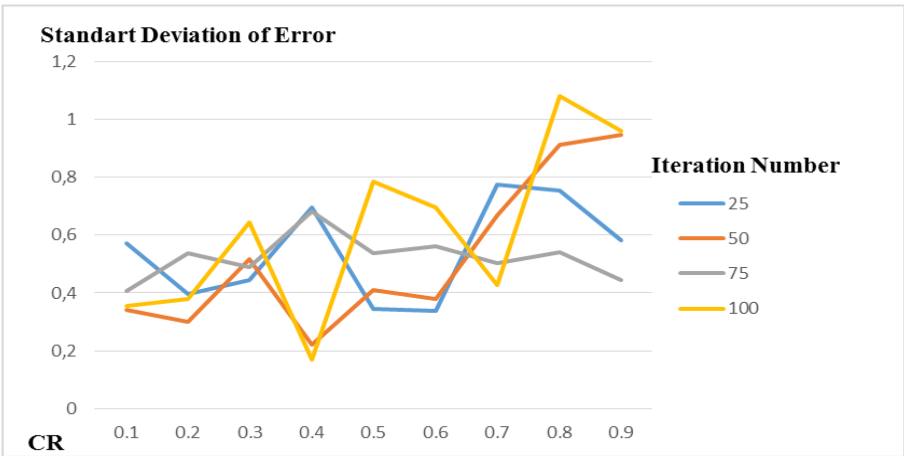

Figure 5.39. STDDEV of n *ten times results of Rand_1_Bin (Baboon)

Figure 5.40 Represents the IOR of n*ten times iterations' results of Rand_1_Bin (Baboon):

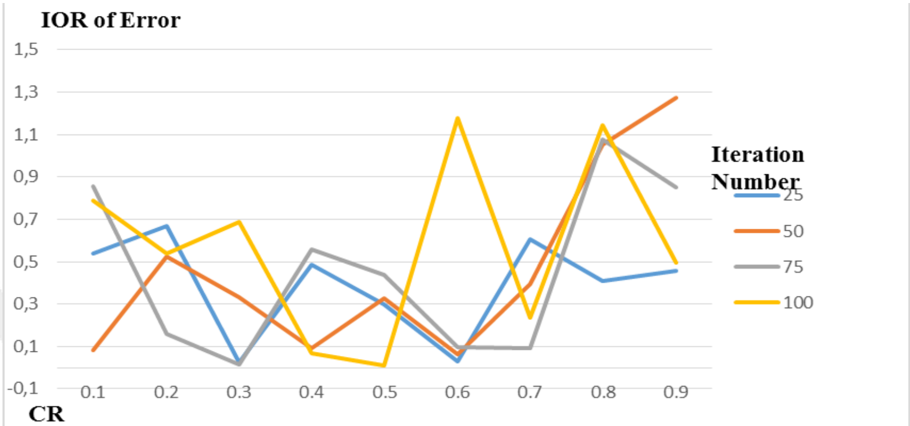

Figure 5.40. IOR of n *ten times iteration results of Rand_1_Bin (Baboon)

Figure 5.41 Represents Original Image Baboon

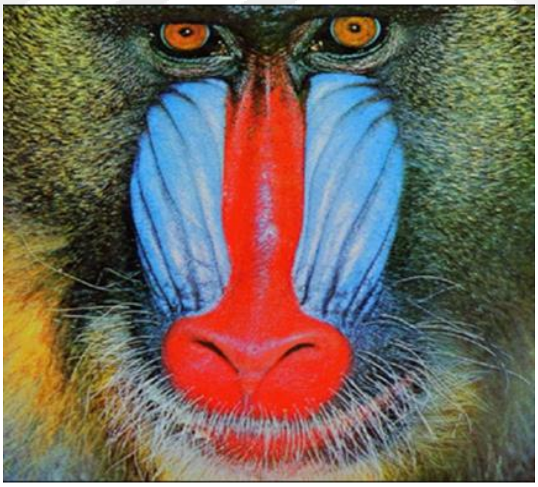

Figure 5.41. Original Image Baboon

Figure 5.42 Represents Rand_1_Bin quantized image at CR: 0.3 25th iteration

Figure 5.42. Rand_1_Bin quantized image at CR: 0.3 25th iteration

5.2.2. Rand_2_Bin Mutation Strategy

Rand_2_Bin mutation strategy number of iterations had repeated ten times then the average , median , standard deviation , interquartile range of these values are calculated.In the graphics error had been divided into 1000 in order to show error in small ranges.

Researches showed that in Rand_2_Bin mutation strategy at CR: 0.3 in 75th iteration at Flowers, CR: 0.1 at 75th iteration at Lenna, CR: 0.2 at 75th iteration at Baboon. Averge, median, standart deviation and interquartile range shows the details in iterations. By average values, it is seen that best values in Rand_2_Bin mutation strategy had reached at CR: 0.2 and 0.8 values at 75 th iterations for Flowers image, CR: 0.2 at 25th iterations for Baboon image and CR: 0.3 at 25th iterations for Lenna image.

Figure 5.43 Represents the n times iterations' results of Rand_2_Bin strategy Flowers:

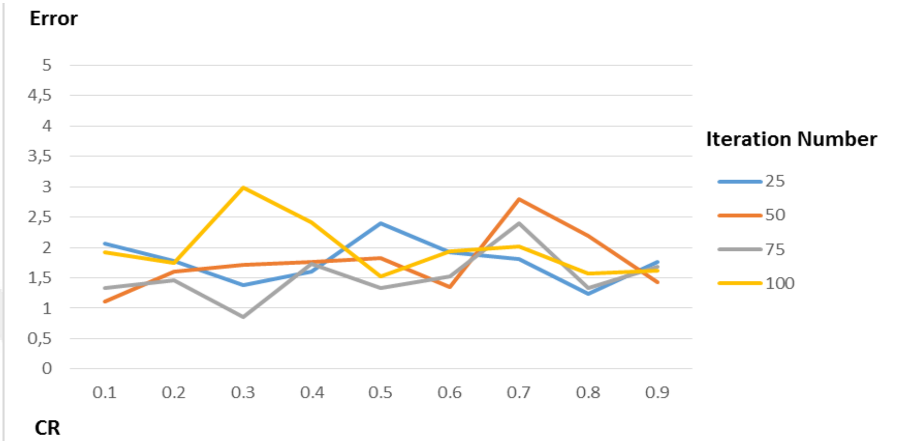

Figure 5.43. n times iterations' results of Rand_2_Bin strategy (Flowers)

Figure 5.44 Represents the average of n*ten times iterations' results of Rand_2_Bin strategy (Flowers):

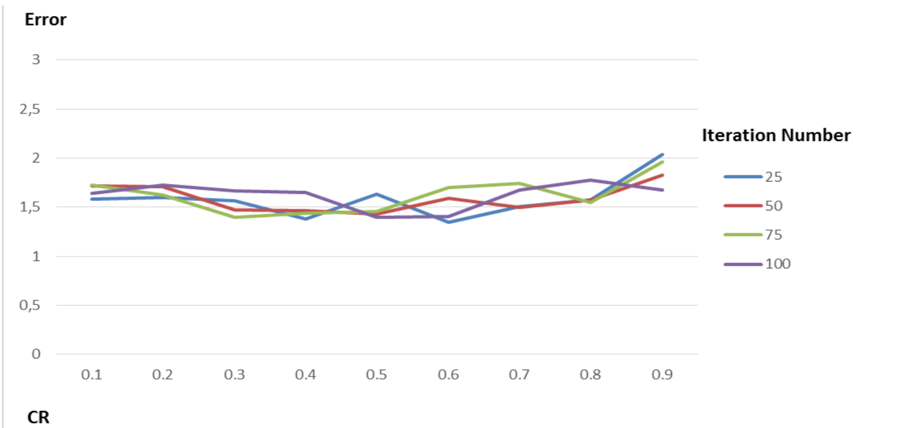

Figure 5.44. Average of n*ten times iterations' results of Rand_2_Bin strategy (Flowers)

Figure 5.45 Represents the median of n*ten times iterations' results of Rand_2_Bin strategy (Flowers):

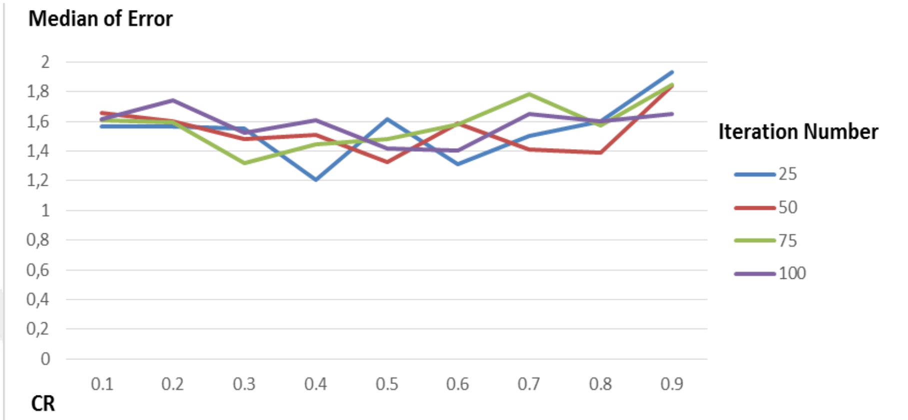

Figure 5.45. Median of n*ten times iterations' results of Rand_2_Bin strategy (Flowers)

Figure 5.46 Represents the STDDEV of n*ten times iterations' results of Rand_2_Bin strategy (Flowers):

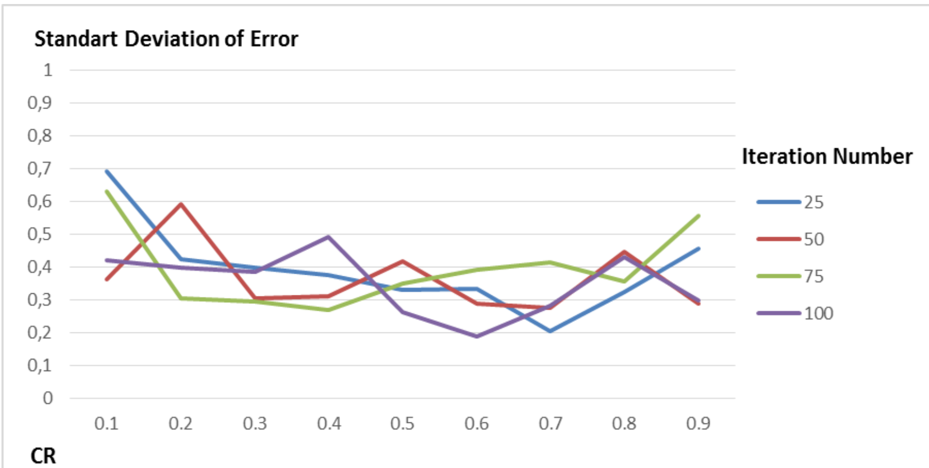

Figure 5.46. STDDEV of n*ten times iterations' results of Rand_2_Bin strategy (Flowers)

Figure 5.47 Represents the IOR of n*ten times iterations' results of Rand_2_Bin strategy (Flowers):

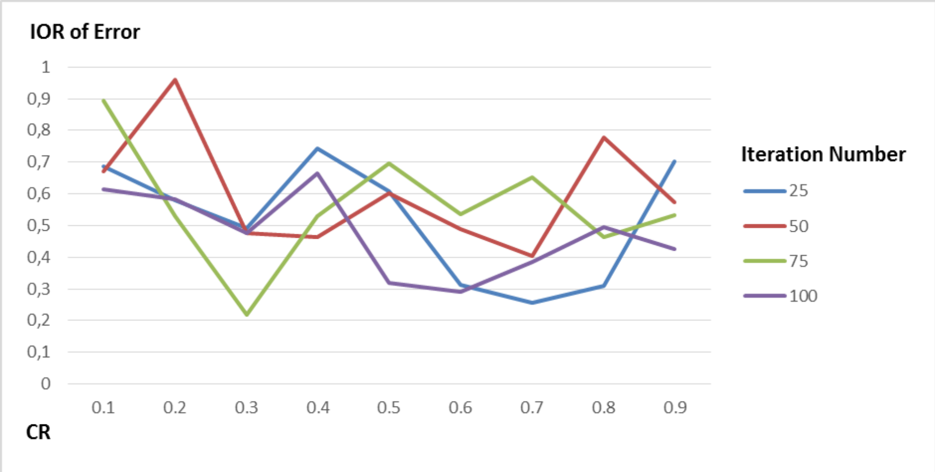

Figure 5.47. IOR of n*ten times iterations' results of Rand_2_Bin strategy (Flowers)

Figure 5.48 represents Original Image Flowers

Figure 5.48. Original Image Flowers

Figure 5.49 represents Rand_2_Bin quantized image CR: 0.3 in 75 iteration

Figure 5.49. Rand_2_Bin quantized image at CR: 0.3 in 75 iteration

Figure 5.50 Represents the n times iterations' results of Rand_2_Bin strategy Lenna:

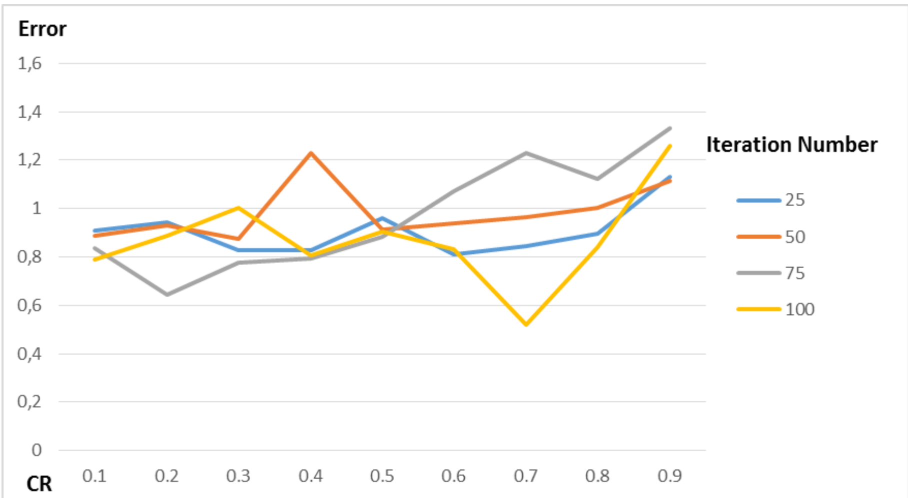

Figure 5. 50. n times iterations' results of Rand_2_Bin strategy (Lenna)

Figure 5.51 represents the average of n*ten times iterations' results of Rand_2_Bin strategy (Lenna)

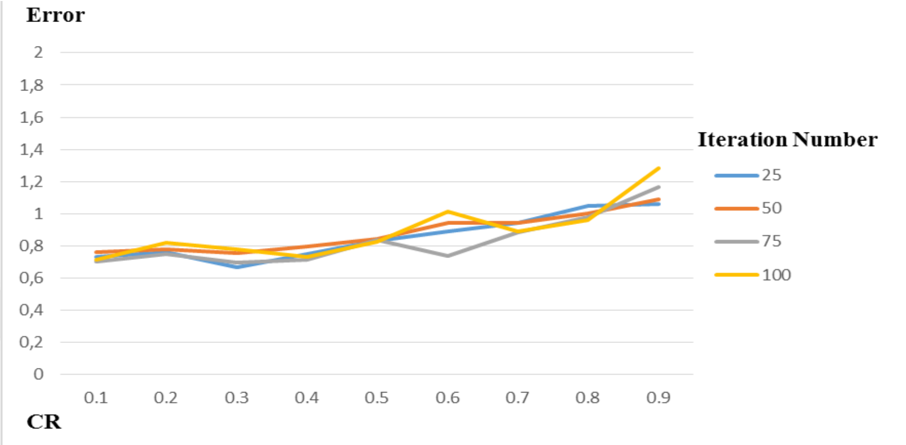

Figure 5.51. Average of n*ten times iterations' results of Rand_2_Bin strategy (Lenna)

Figure 5.52 represents the median of n*ten times iterations' results of Rand_2_Bin strategy (Lenna)

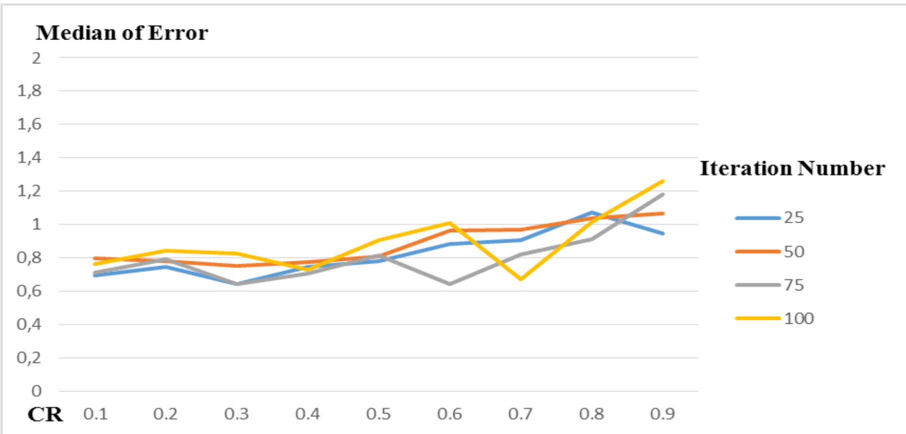

Figure 5.52. Median of n*ten times iterations' results of Rand_2_Bin strategy (Lenna)

Figure 5.53 represents the STDDEV of n*ten times iterations' results of Rand_2_Bin strategy (Lenna)

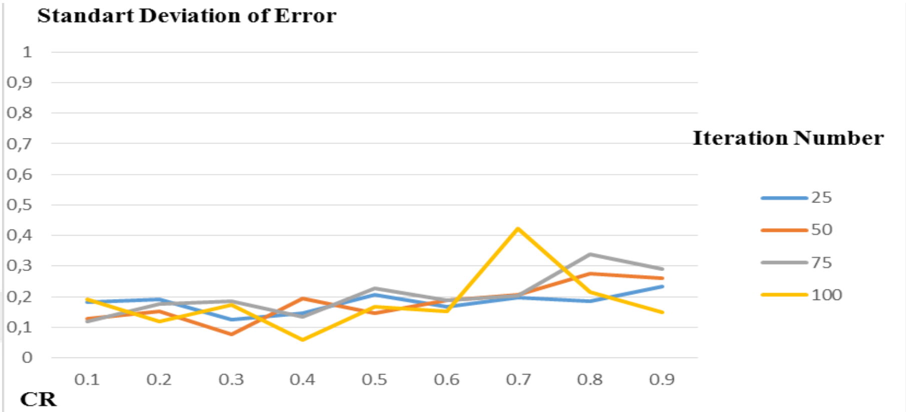

Figure 5.53. STDDEV of n*ten times iterations' results of Rand_2_Bin strategy (Lenna)

Figure 5.54 represents the IOR of n*ten times iterations' results of Rand_2_Bin strategy (Lenna)

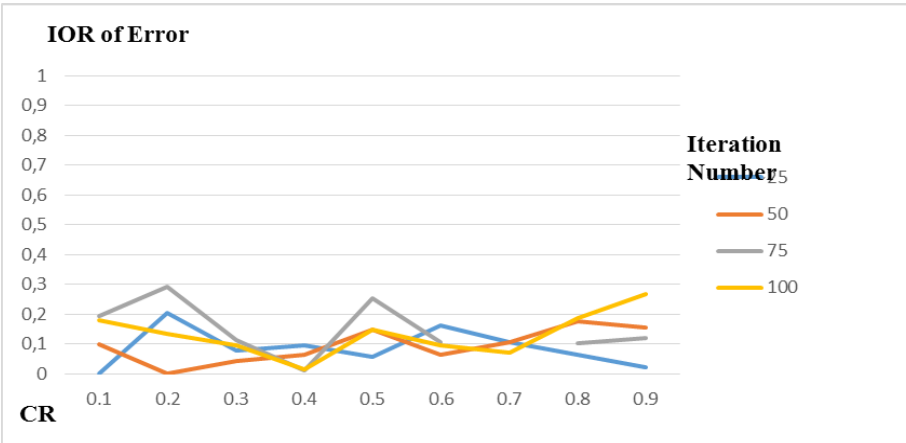

Figure 5.54. IOR of n*ten times iterations' results of Rand_2_Bin strategy (Lenna)

Figure 5.55 represents Original Image Lenna

Figure 5.55. Original Image Lenna

Figure 5.56 represents Rand_2_Bin quantized image at CR: 0.3 in 25th iteration

Figure 5. 56. Rand_2_Bin quantized image at CR: 0.3 in 25th iteration

Figure 5.57 represents the n times iterations' results of Rand_2_Bin strategy Baboon:

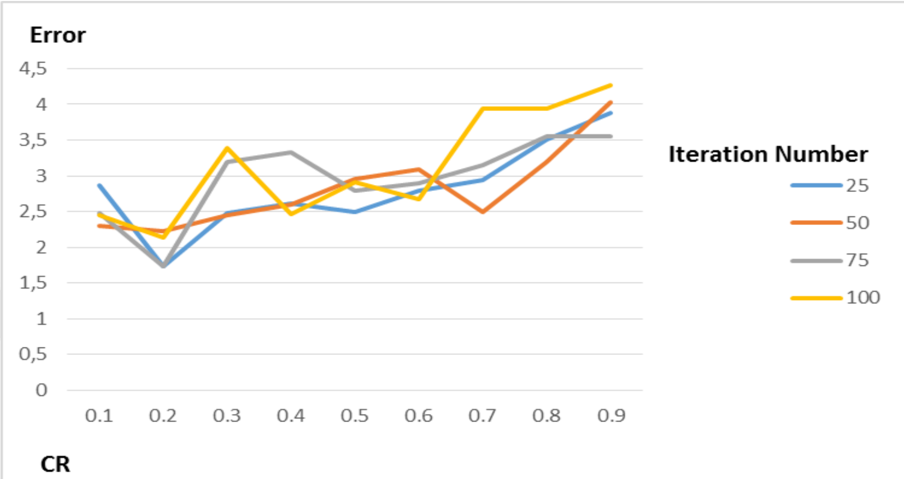

Figure 5.57**.** n times iterations' results of Rand_2_Bin strategy (Baboon)

Figure 5.58 represents the average of n*ten times iterations' results of Rand_2_Bin strategy (Baboon)

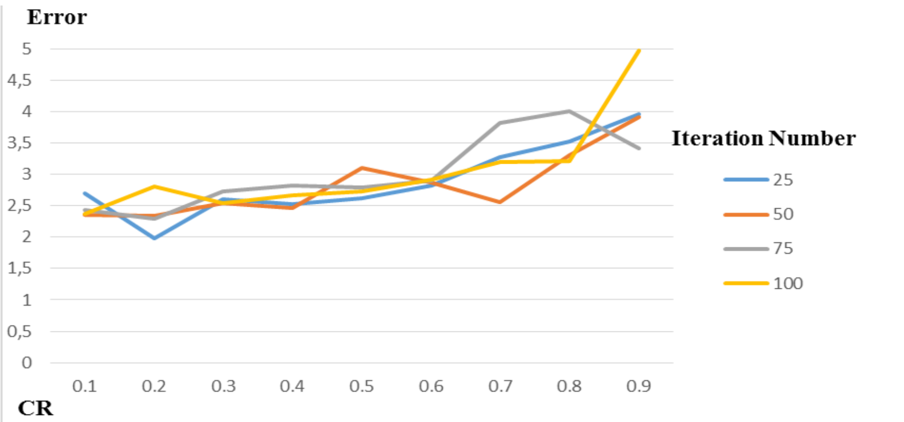

Figure 5.58. Average of n*ten times iterations' results of Rand_2_Bin strategy (Baboon)

Figure 5.59 represents the median of n*ten times iterations' results of Rand_2_Bin strategy (Baboon)

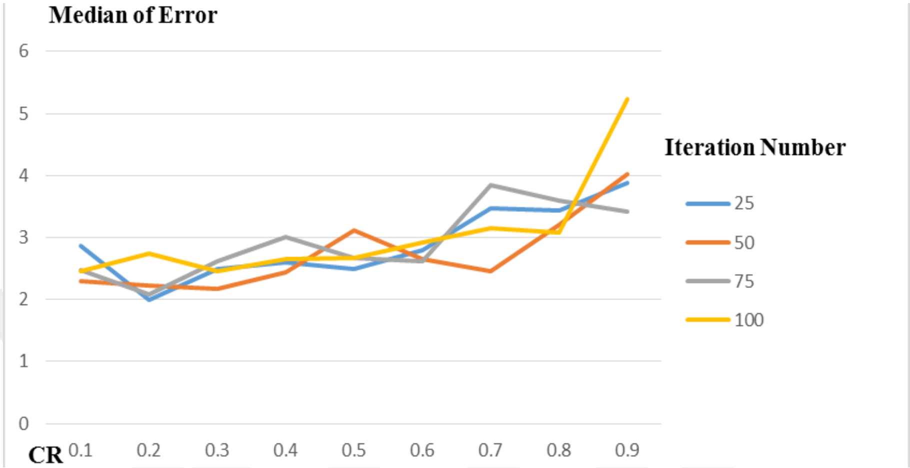

Figure 5.59. Median of n*ten times iterations' results of Rand_2_Bin strategy (Baboon)

Figure 5.60 Represents the STDDEV of n*ten times iterations' results of Rand_2_Bin strategy (Baboon)

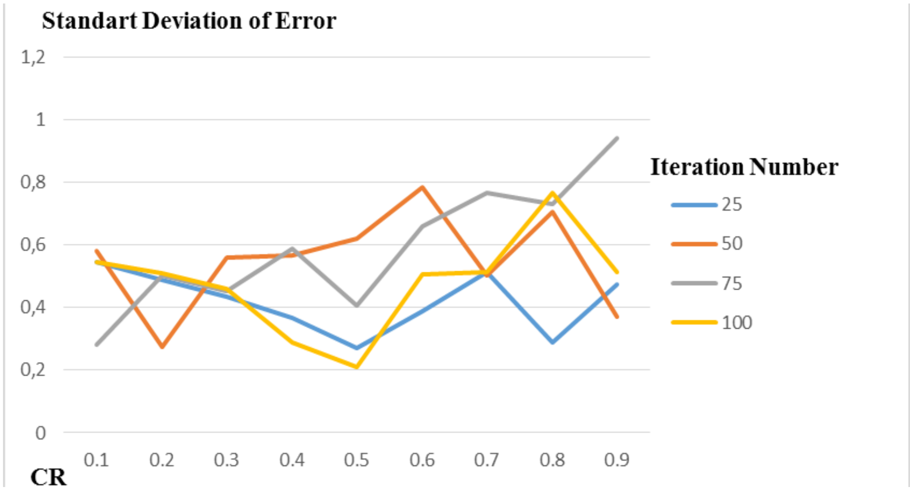

Figure 5.60. STDDEV of n*ten times iterations' results of Rand_2_Bin strategy (Baboon)

Figure 5.61 Represents the IOR of n*ten times iterations' results of Rand_2_Bin strategy (Baboon) :

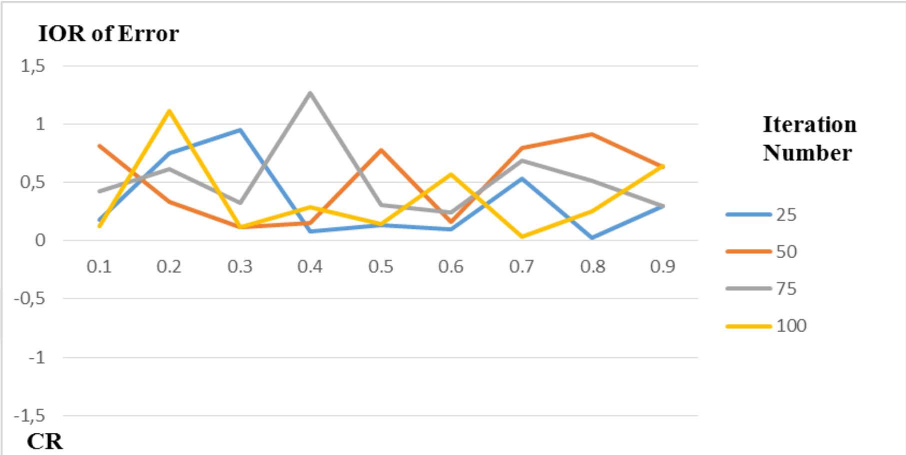

Figure 5.61. IOR of n*ten times iterations' results of Rand_2_Bin strategy (Baboon)

Figure 5.64 represents Original Image Baboon

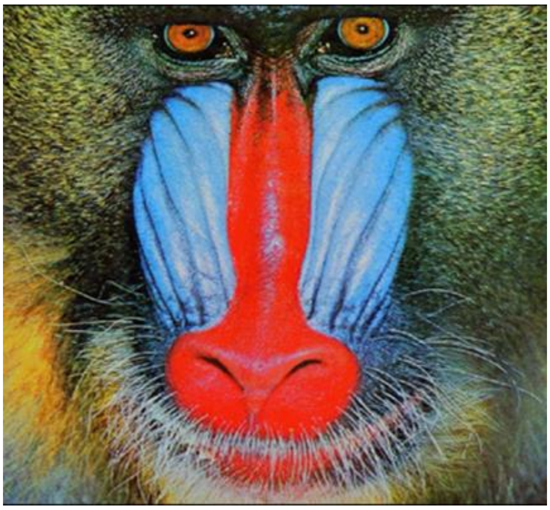

Figure 5.62. Original Image Baboon

Figure 5.63 represents Rand_2_Bin quantized image at CR: 0.2 in 25 th iteration

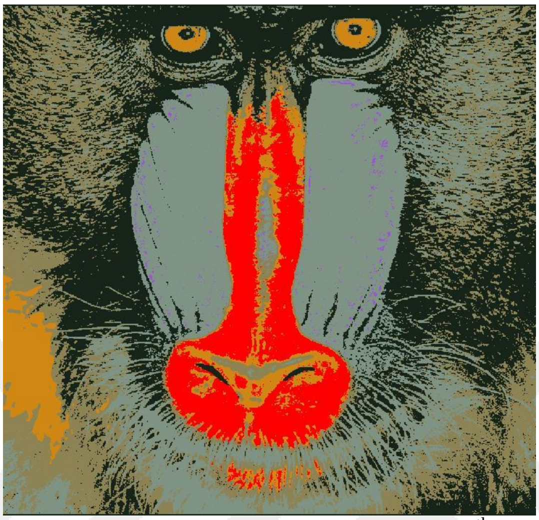

Figure 5.63**.** Rand_2_Bin quantized image at CR: 0.2 in 25 th iteration

5.2.3. Best_1_Bin Mutation Strategy

Best_1_Bin mutation strategy number of iterations had repeated ten times then the average , median , standard deviation , interquartile range of these values are calculated.In the graphics error had been divided into 1000 in order to show error in small ranges.

Researches showed that in Best_1_Bin mutation strategy at CR: 0.2 in 50th iteration at Flowers, CR: 0.3 at 25th iteration at Lenna, CR: 0.1 at 25th iteration at Baboon. Averge, median, standart deviation and interquartile range shows the details in iterations. By average values, it is seen that best values in Best_1_Bin mutation strategy had reached at CR: 0.2 and 0.6 values at 25 th iterations for Flowers image, CR: 0.1 at 25th iterations for Baboon image and CR: 0.2 at 100th iterations for Lenna image.

Figure 5.64 represents the results of n times iterations of Best_1_Bin strategy (Flowers):

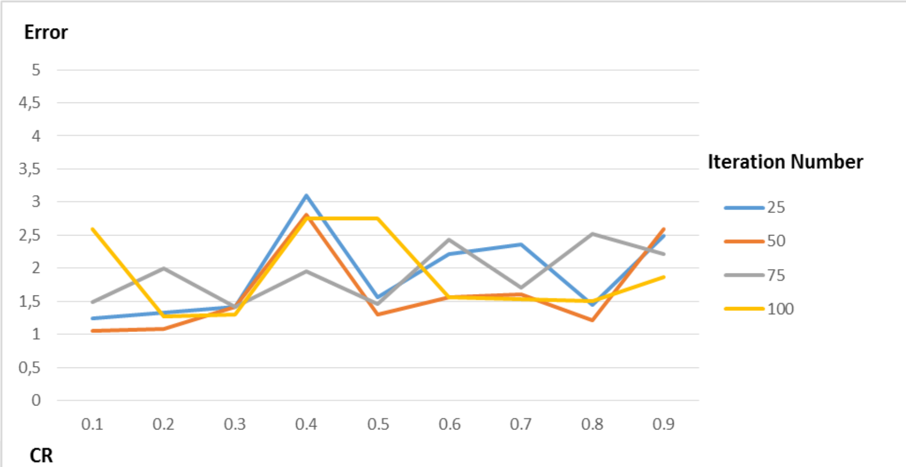

Figure 5.64. n times iterations of Best_1_Bin strategy (Flowers)

Figure 5.65 represents the average of n*ten times iterations' results of Best_1_Bin strategy (Flowers):

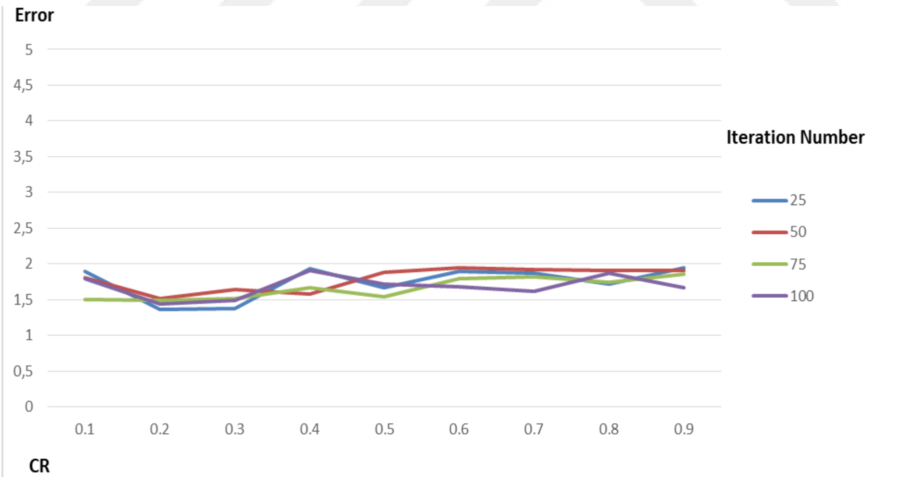

Figure 5. 65. Average of n*ten times iterations' results of Best_1_Bin strategy (Flowers)

Figure 5.66 represents the median of n*ten times iterations' results of Best_1_Bin strategy:

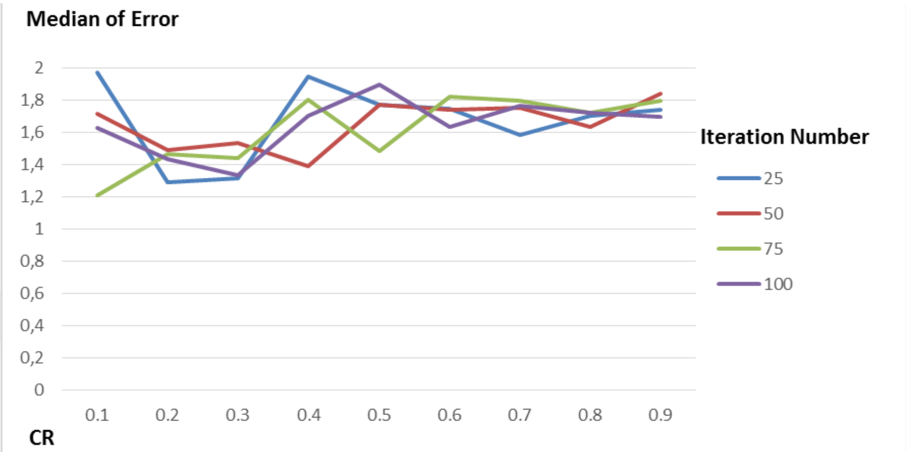

Figure 5.66. Median of n*ten times iterations' results of Best_1_Bin strategy (Flowers)

Figure 5.70 represents the STDDEV of n*ten times iterations' results of Best_1_Bin strategy (Flowers):

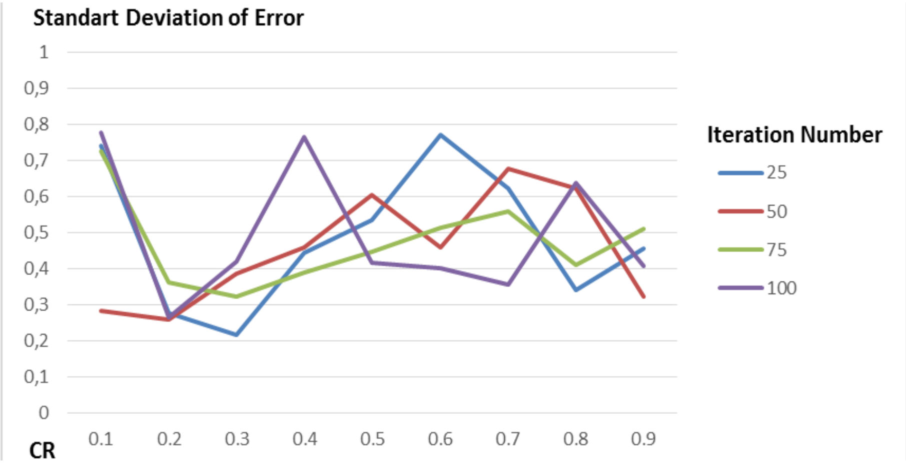

Figure 5.67. STDDEV of of n*ten times iterations' results of Best_1_Bin strategy (Flowers)

Figure 5.68 represents the IOR of of n*ten times iterations' results of Best_1_Bin strategy (Flowers):

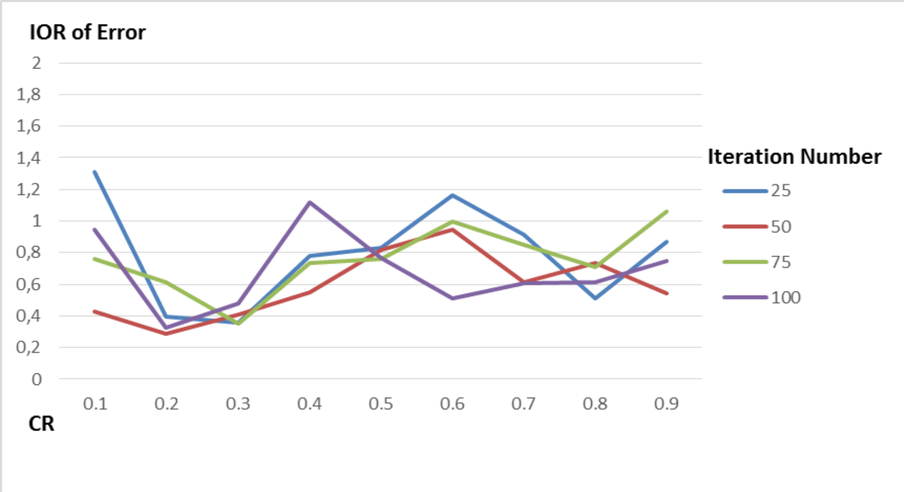

Figure 5. 68. IOR of n*ten times iterations' results of Best_1_Bin strategy (Flowers)

Figure 5.69 represents Original Image Flowers

Figure 5.69. Original Image Flowers

Figure 5.70 represents the quantized image by Best_1_Bin strategy at CR: 0.2 in 50th iteration:

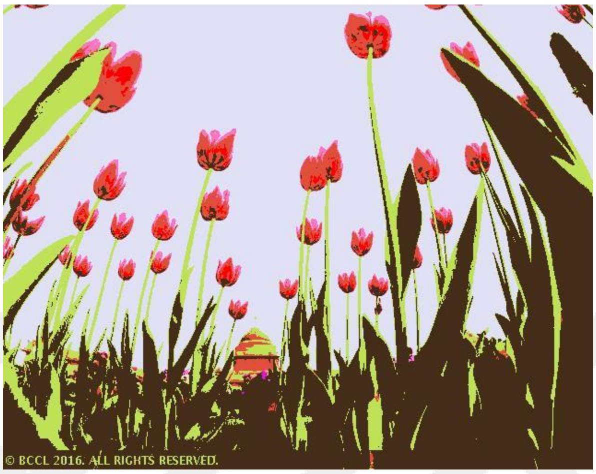

Figure 5.70. quantized image by Best_1_Bin strategy at CR: 0.2 in 50th iteration

Figure 5.71 represents the results of n times iterations of Best_1_Bin strategy (Lenna):

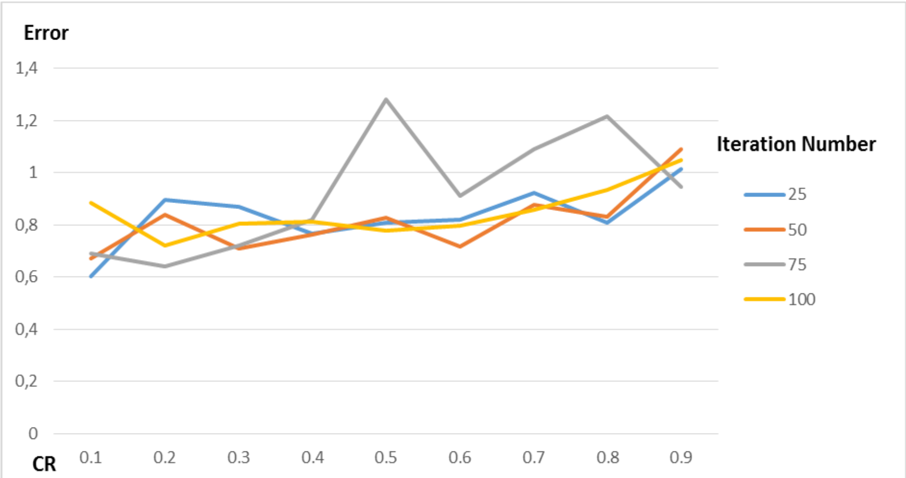

Figure 5.71. n times iterations of Best_1_Bin strategy (Lenna)

Figure 5.72 represents the average of n*ten times iterations' results of Best_1_Bin strategy (Lenna):

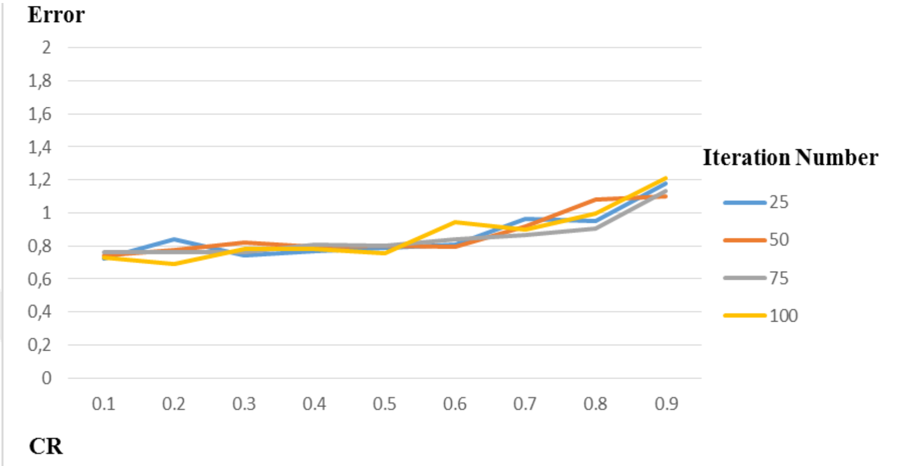

Figure 5.72. Average of n*ten times iterations' results of Best_1_Bin strategy (Lenna)

Figure 5.73 represents the median of n*ten times iterations' results of Best_1_Bin strategy:

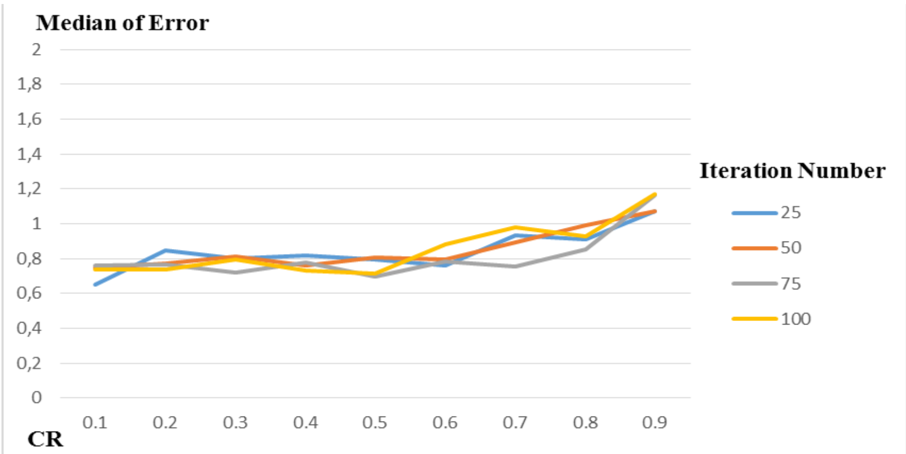

Figure 5.73**.** Median of n*ten times iterations' results of Best_1_Bin strategy (Lenna)

Figure 5.74 represents the STDDEV of n*ten times iterations' results of Best_1_Bin strategy (Lenna):

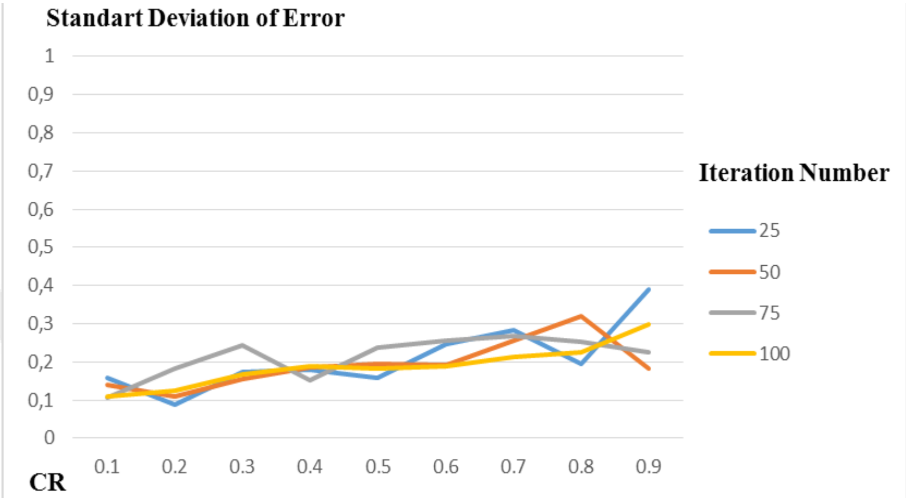

Figure 5.74. STDDEV of of n*ten times iterations' results of Best_1_Bin strategy (Lenna)

Figure 5.75 represents the IOR of of n*ten times iterations' results of Best_1_Bin strategy (Lenna):

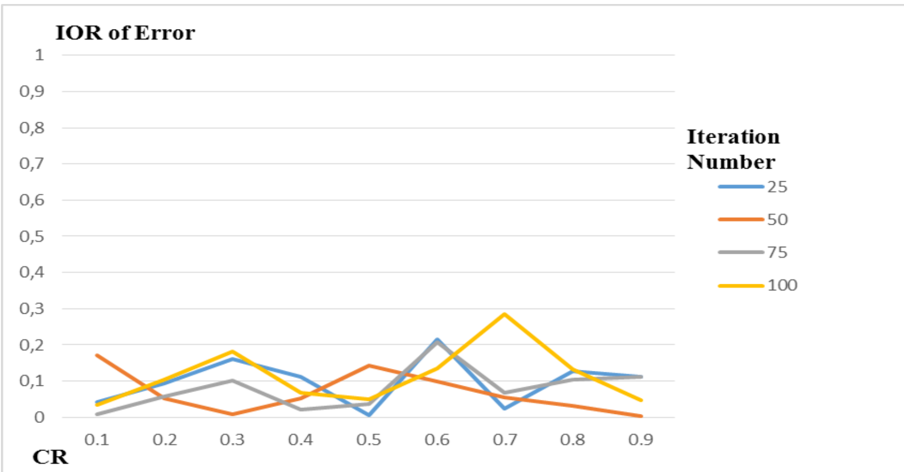

Figure 5.75. IOR of n*ten times iterations' results of Best_1_Bin strategy (Lenna)

Figure 5.76 Represents Original Image Lenna

Figure 5.76. Original Image Lenna

Figure 5.77 represents the quantized image by Best_1_Bin strategy at CR: 0.1 in 25th iteration:

Figure 5.77. Quantized image by Best_1_Bin strategy at CR: 0.1 in 25th iteration

Figure 5.78 represents the results of n times iterations of Best_1_Bin strategy (Baboon):

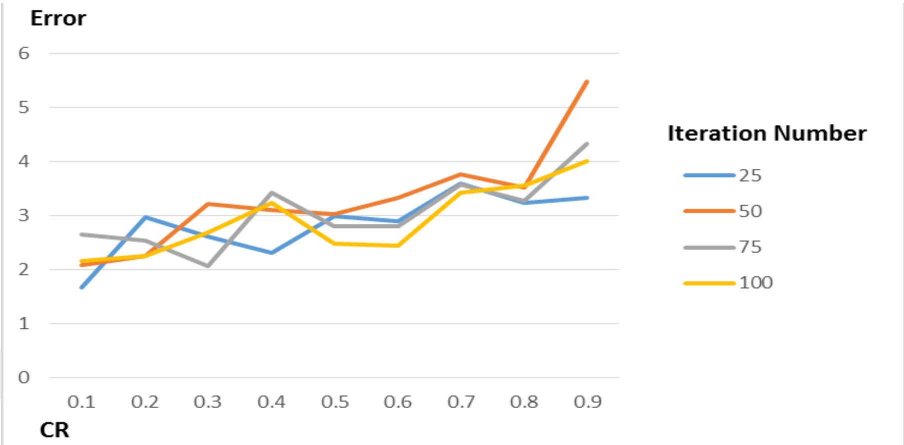

Figure 5.78n times iterations of Best_1_Bin strategy (Baboon)

Figure 5.79 represents the average of n*ten times iterations' results of Best_1_Bin strategy (Baboon):

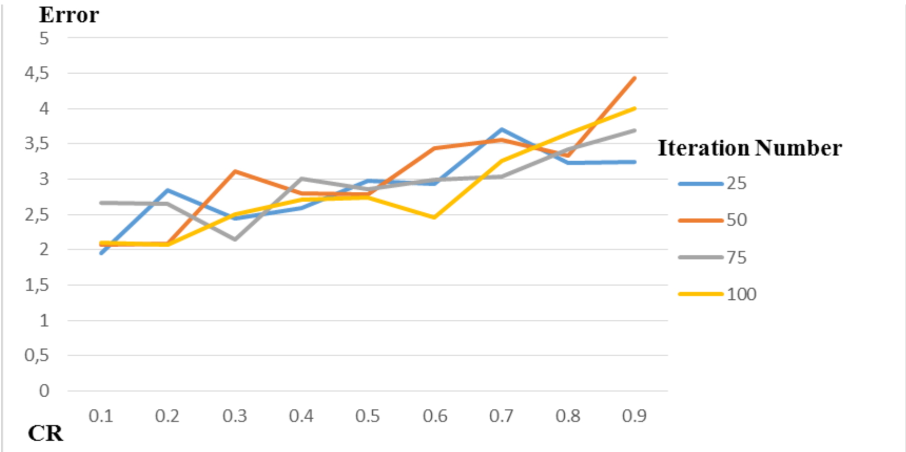

Figure 5.79. Average of n*ten times iterations' results of Best_1_Bin strategy (Baboon)

Figure 5.80 represents the median of n*ten times iterations' results of Best_1_Bin strategy (Baboon):

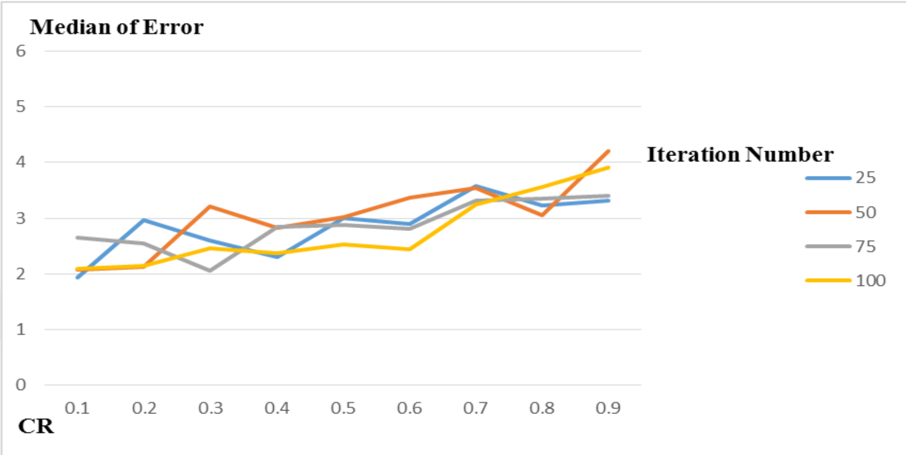

Figure 5. 80. Median of n*ten times iterations' results of Best_1_Bin strategy (Baboon)

Figure 5.81 represents the STDDEV of n*ten times iterations' results of Best_1_Bin strategy (Baboon):

Figure 5. 81. STDDEV of of n*ten times iterations' results of Best_1_Bin strategy (Baboon)

Figure 5.82 represents the IOR of of n*ten times iterations' results of Best_1_Bin strategy (Baboon):

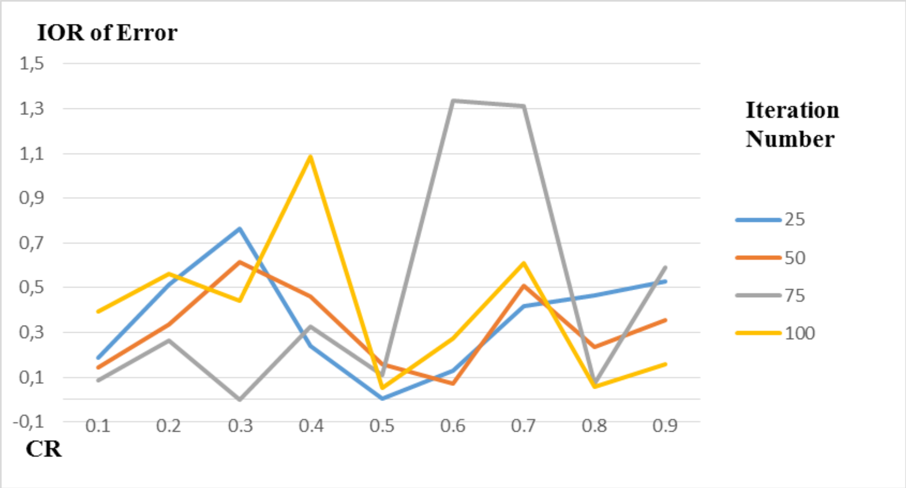

Figure 5. 82. IOR of n*ten times iterations' results of Best_1_Bin strategy (Baboon)

Figure 5.83 represents Original Image Baboon

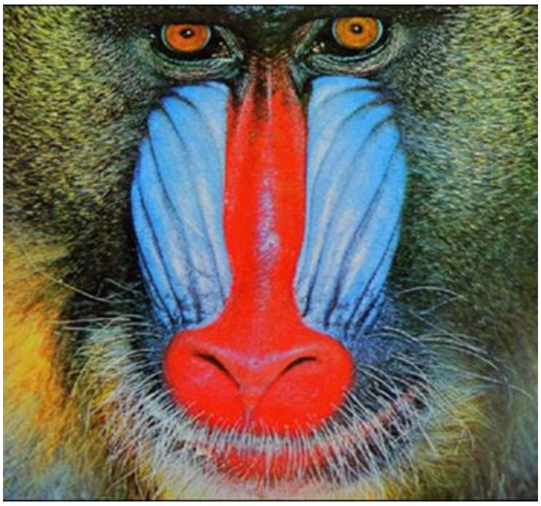

Figure 5.83. Original Image Baboon

Figure 5.84 represents the quantized image by Best_1_Bin strategy at CR: 0.1 in 25th iteration:

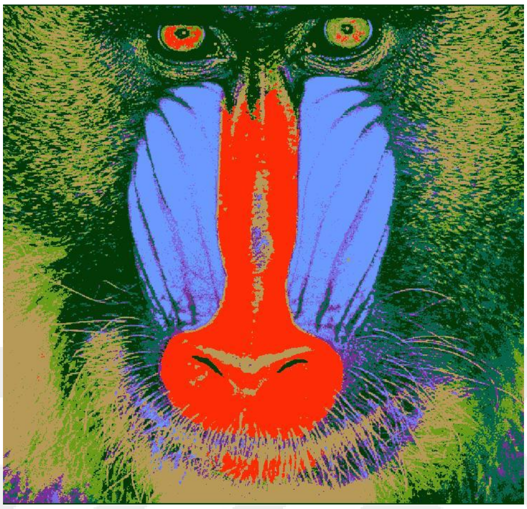

Figure 5.84. quantized image by Best_1_Bin strategy at CR: 0.1 in 25th iteration

5.2.4. Best_2_Bin Mutation Strategy

Best_2_Bin mutation strategy number of iterations had repeated ten times then the average , median , standard deviation , interquartile range of these values are calculated.In the graphics error had been divided into 1000 in order to show error in small ranges.

Researches showed that in Best_2_Bin mutation strategy at CR: 0.4 in 50th iteration at Flowers, CR: 0.1 at 25th iteration at Lenna, CR: 0.1 at 50th iteration at Baboon. Averge, median, standart deviation and interquartile range shows the details in iterations. By average values, it is seen that best values in Best_1_Bin mutation strategy had reached at CR: 0.3 and 0.6 values at 50 th iterations for Flowers image, CR: 0.2 at 100th iterations for Baboon image and CR: 0.2 at 100th iterations for Lenna image.

Figure 5.85 represents n times iterations' results of Best_2_Bin strategy (Flowers) :

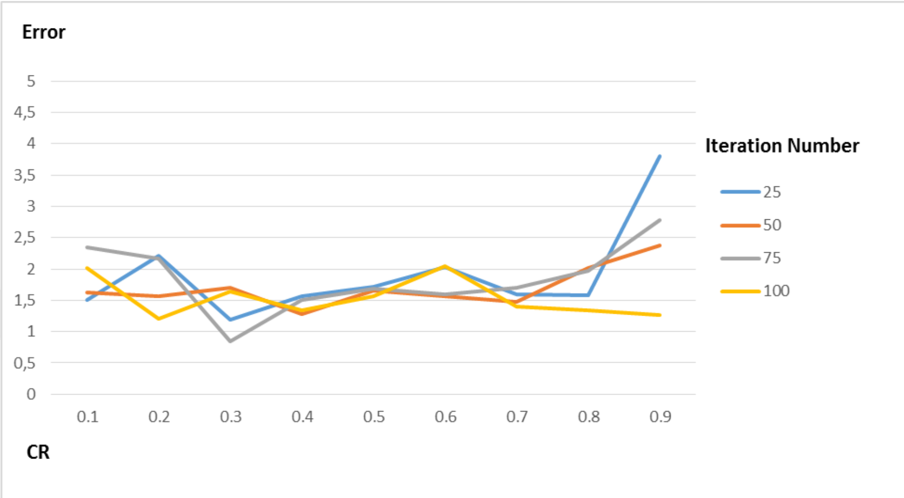

Figure 5. 85. n times iterations' results of Best_2_Bin strategy (Flowers)

Figure 5.86 represents the average of n*ten times iterations' results of Best_2_Bin strategy (Flowers) :

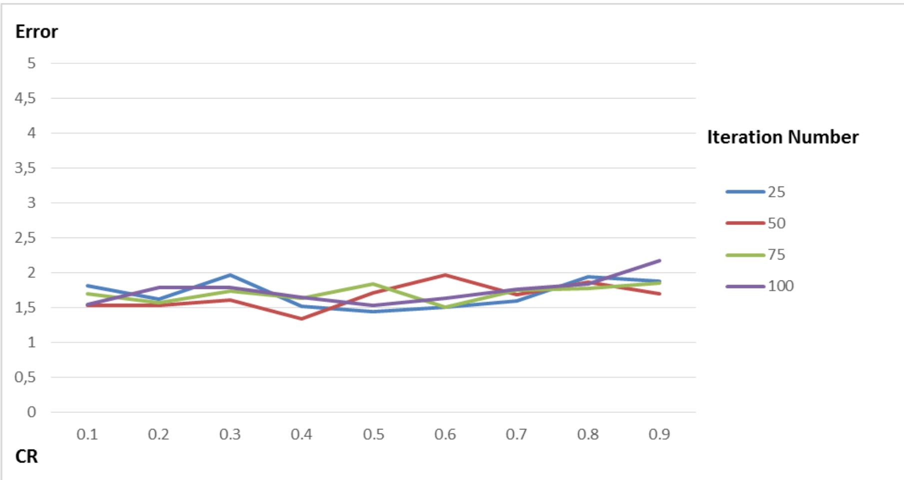

Figure 5. 86. Average of n*ten times iterations' results of Best_2_Bin strategy (Flowers)

Figure 5.87 represents the median of n*ten times iterations' results of Best_2_Bin strategy (Flowers) :

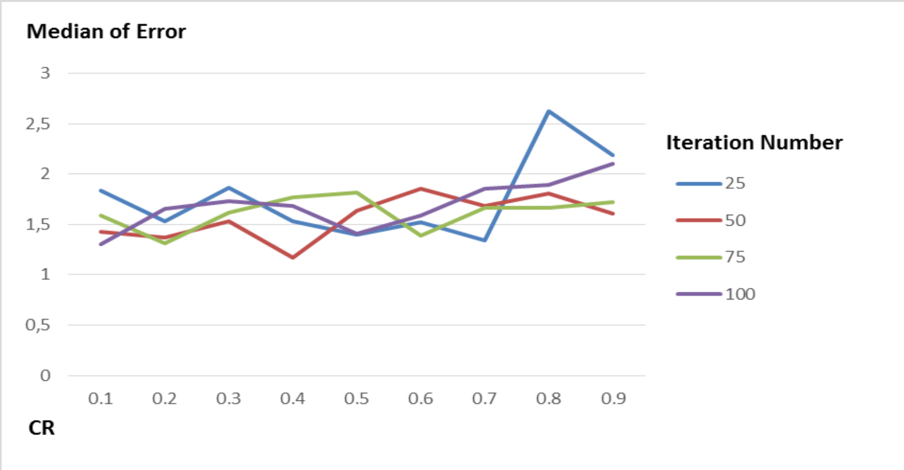

Figure 5.87. Median of n*ten times iterations' results of Best_2_Bin strategy (Flowers)

Figure 5.88 represents the STDDEV of n*ten times iterations' results of Best_2_Bin strategy (Flowers) :

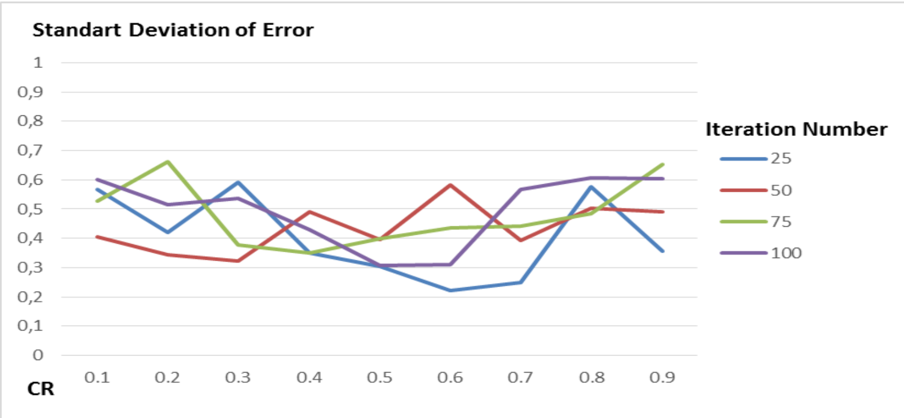

Figure 5. 88. STDDEV of n*ten times iterations' results of Best_2_Bin strategy (Flowers)

Figure 5.89 represents the IOR of n*ten times iterations' results of Best_2_Bin (Flowers) :

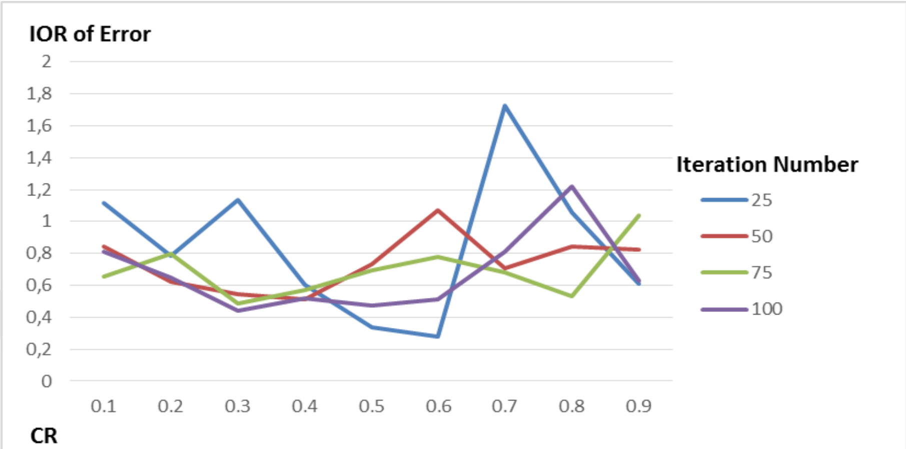

Figure 5. 89. IOR of n*ten times iterations' results of Best_2_Bin (Flowers)

Figure 5.90 represents Original Image Flowers

Figure 5.90. Original Image Flowers

Figure 5.91 represents the quantized image by Best_2_Bin strategy at CR: 0.4 at 50th iteration.

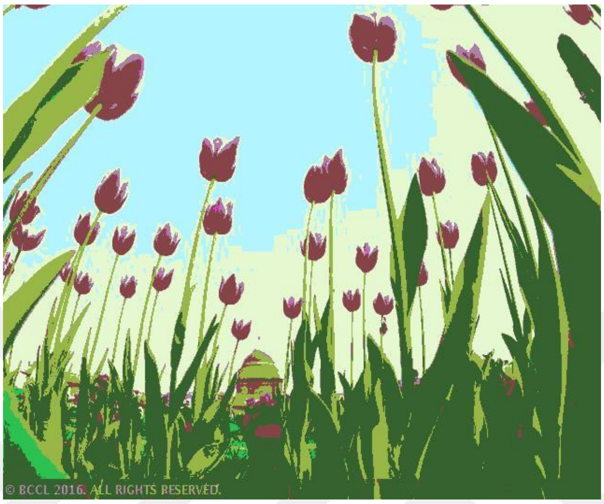

Figure 5. 91. Quantized image by Best_2_Bin strategy at CR :0.4 at 50th iteration

Figure 5.92 represents n times iterations' results of Best_2_Bin strategy (Lenna) :

Figure 5. 92. n times iterations' results of Best_2_Bin strategy (Lenna)

Figure 5. 93 represents the average of n*ten times iterations' results of Best_2_Bin strategy (Lenna) :

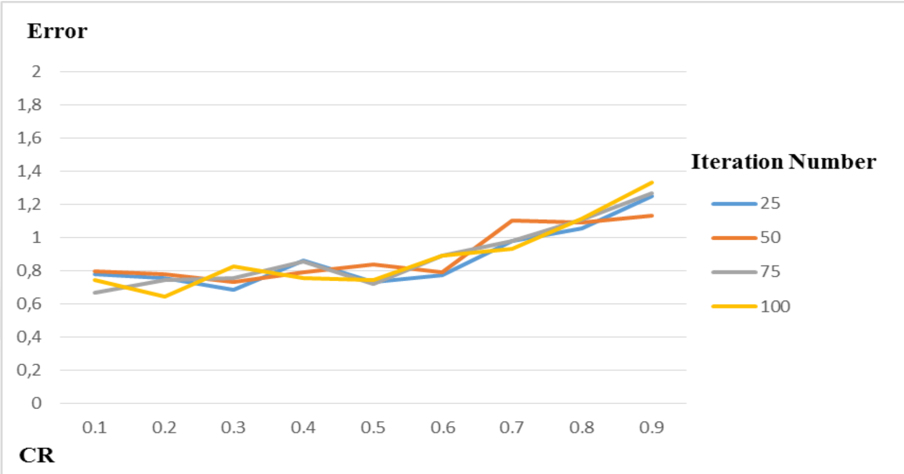

Figure 5.93. Average of n*ten times iterations' results of Best_2_Bin strategy (Lenna)

Figure 5.94 represents the median of n*ten times iterations' results of Best_2_Bin strategy (Lenna) :

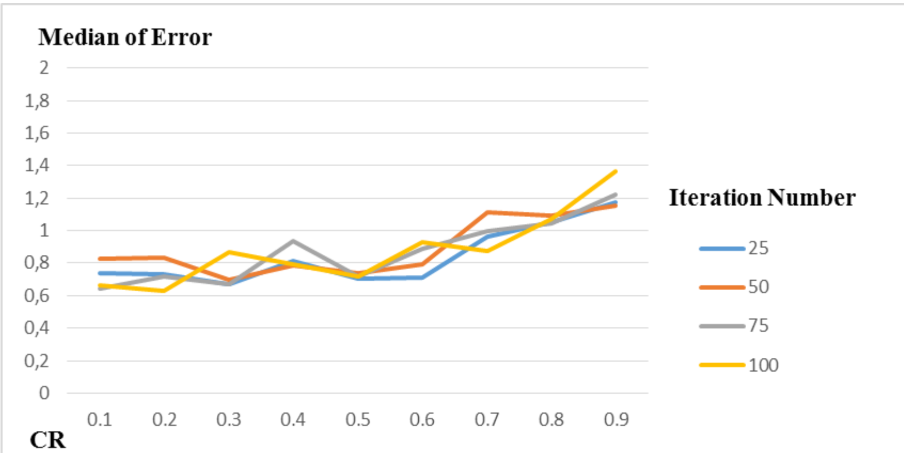

Figure 5. 94. Median of n*ten times iterations' results of Best_2_Bin strategy (Lenna)

Figure 5.95 represents the STDDEV of n*ten times iterations' results of Best_2_Bin strategy (Lenna):

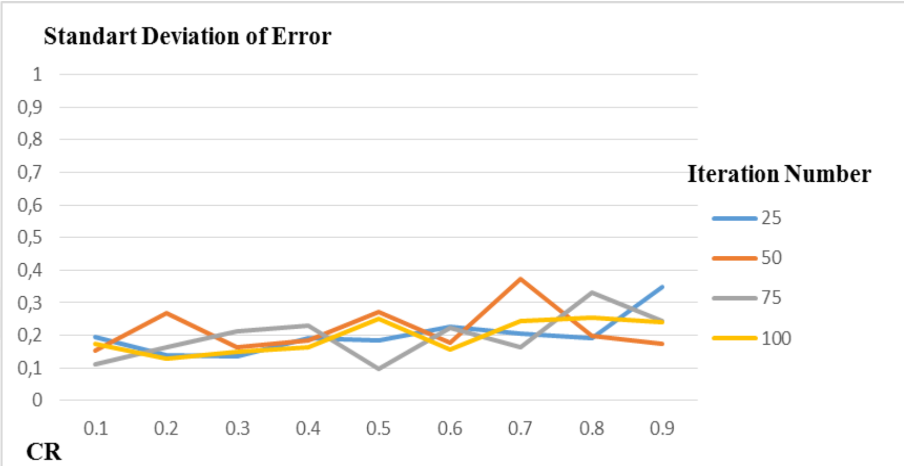

Figure 5.95. STDDEV of n*ten times iterations' results of Best_2_Bin strategy (Lenna)

Figure 5.96 represents the IOR of n*ten times iterations' results of Best_2_Bin (Lenna)

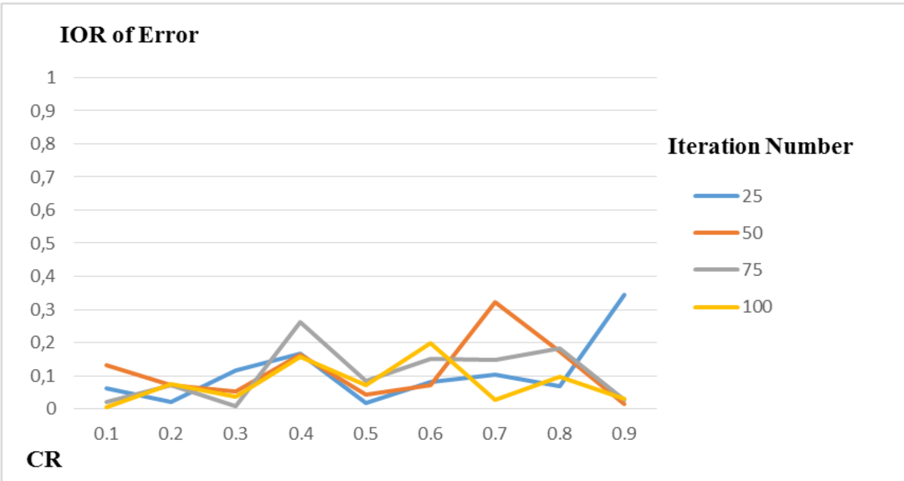

Figure 5.96. IOR of n*ten times iterations' results of Best_2_Bin (Lenna)

Figure 5.97 represents Original Image Lenna

Figure 5.97**.** Original Image Lenna

Figure 5.98 represents the quantized image by Best_2_Bin strategy at CR: 0.1 at 25th iteration

Figure 5.98**.** Quantized image by Best_2_Bin strategy at CR: 0.1 at 25th iteration

Figure 5.99 represents n times iterations' results of Best_2_Bin strategy (Baboon) :

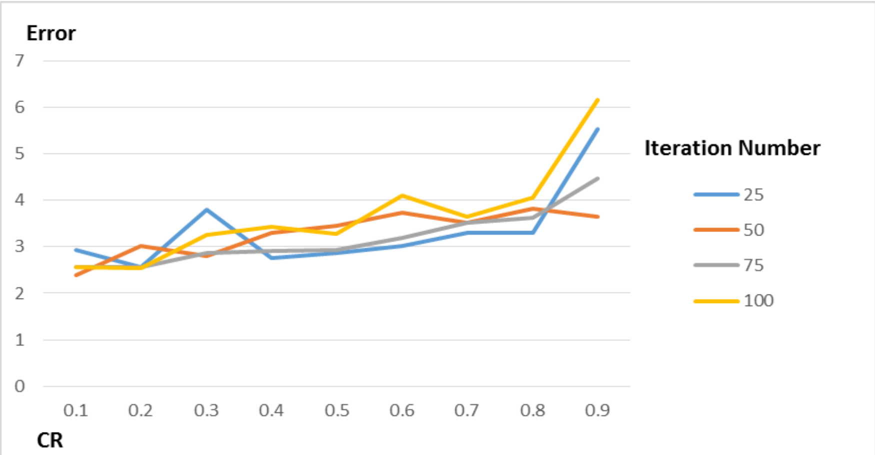

Figure 5. 99. n times iterations' results of Best_2_Bin strategy (Baboon)

Figure 5.100 represents the average of n*ten times iterations' results of Best_2_Bin strategy (Baboon) :

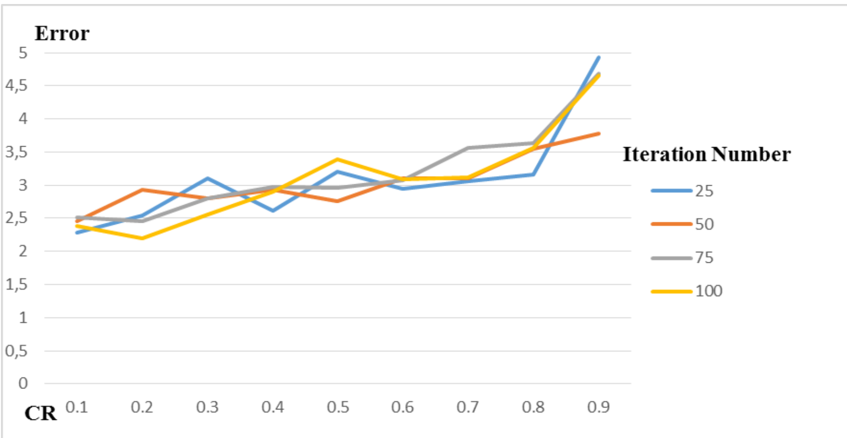

Figure 5.100. Average of n*ten times iterations' results of Best_2_Bin strategy (Baboon)

Figure 5.101 represents the median of n*ten times iterations' results of Best_2_Bin strategy (Baboon) :

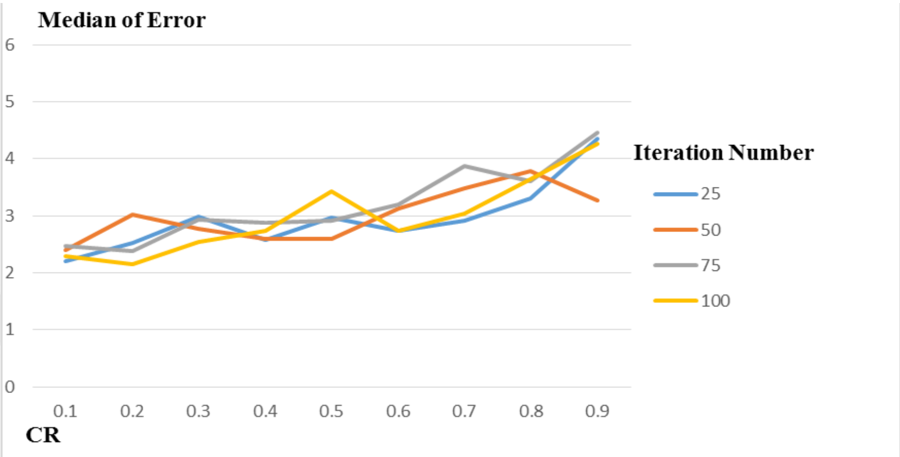

Figure 5. 101. Median of n*ten times iterations' results of Best_2_Bin strategy (Baboon)

Figure 5.102 represents the STDDEV of n*ten times iterations' results of Best_2_Bin strategy (Baboon) :

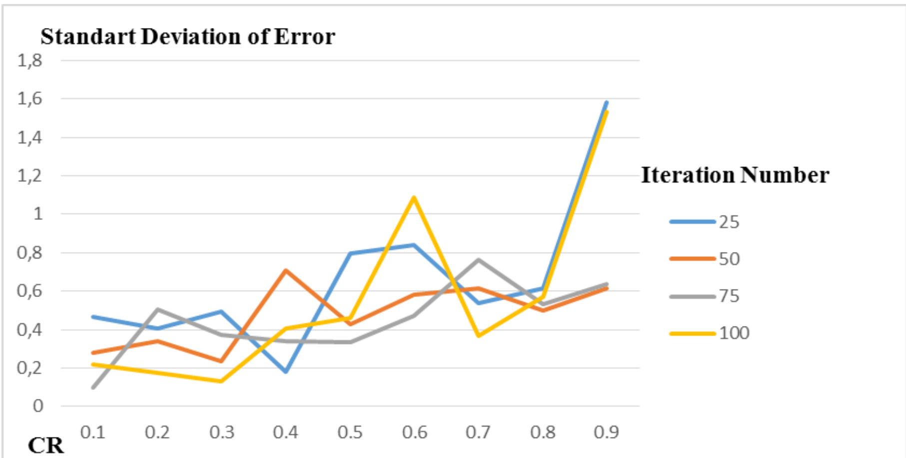

Figure 5. 102. STDDEV of n*ten times iterations' results of Best_2_Bin strategy (Baboon)

Figure 5.103 represents the IOR of n*ten times iterations' results of Best_2_Bin (Baboon) :

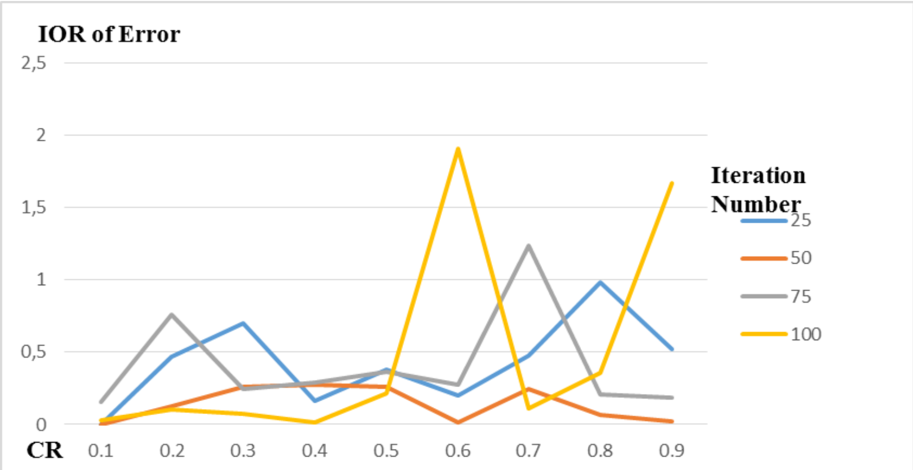

Figure 5.103. IOR of n*ten times iterations' results of Best_2_Bin (Baboon)

Figure 5.104 represents Original Image Baboon

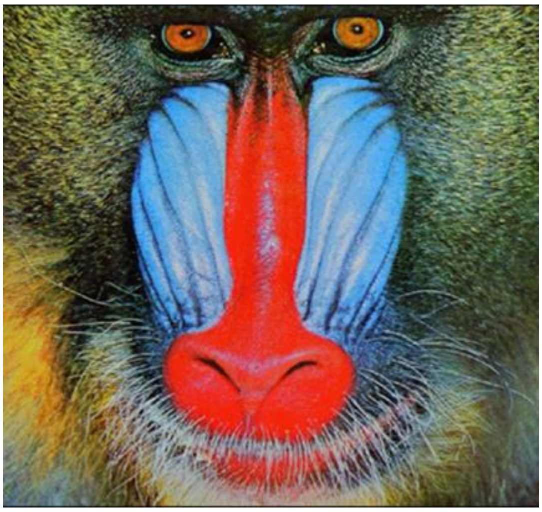

Figure 5.104. Original Image Baboon

Figure 5.105 represents the quantized image by Best_2_Bin strategy at CR: 0.1 at 25^{th} iteration

Figure 5.105**.** Quantized image by Best_2_Bin strategy at CR: 0.1 at 25^{th} iteration

5.2.5. Comparisions of Mutation Strategies

In color quantizing by DE Best_2_Bin Mutation strategy had reached the least error in n*ten times iterations.According to the survey in DE algorithm Best_2_Bin had been the fastest and had had least error palette.The reliability of results in whole strategies is tested. Results had been approximetly.Here is the results table of all strategies.

Tables 5.1, 5.2, 5.3 show the results of all Mutation Strategies of Differential Evolution Algorithm.

Table 5.1 Mutation Strategies of Differential Evolution Algorithm (Flowers):

Mutation Strategy	Iteration Number	CR	ERROR
Rand_1_Bin	50	0.2	1.38
Rand_2_Bin	75	0.3	1.39
Best_1_Bin	25	0.2	1.36
Best_2_Bin	50	0.4	1.33

Table 5.2 Mutation Strategies of Differential Evolution Algorithm (Lenna):

Mutation Strategy	Iteration Number	CR	ERROR
Rand_1_Bin	100	0.2	0.85
Rand_2_Bin	25	0.3	0.66
Best_1_Bin	100	0.2	0.69
Best_2_Bin	100	0.2	0.64

Table 5.3 Mutation Strategies of Differential Evolution Algorithm (Baboon):

Mutation Strategy	Iteration Number	CR	ERROR
Rand_1_Bin	100	0.2	2.34
Rand_2_Bin	25	0.2	1.98
Best_1_Bin	25	0.1	1.94
Best_2_Bin	100	0.2	2.20

5.3. Results of Color Quantizing Based on GWO

GWO algorithm is a meta-heuristic algorithm that means the first initial values are started by random values. Because of random initialization step, in order to check the reliability of program each n number iteration is repeated n*10 times again.

After these result the average of n*10 times iterations errors are compared by the n times iterations. In the representation of graphics error of palettes are divided into 1000 in order to show error in small range.Results had proofed that in two calculations error is decreased while partical of numbers and iterations are increased.

The advantage of GWO had been least error between the image and quantized image. The disadvantage of GWO had been time consuming. Because of this disadvantage at eighty particals results can be showed.

Figure 5.106 shows the results of GWO color quantizing n times iterations (Flowers):

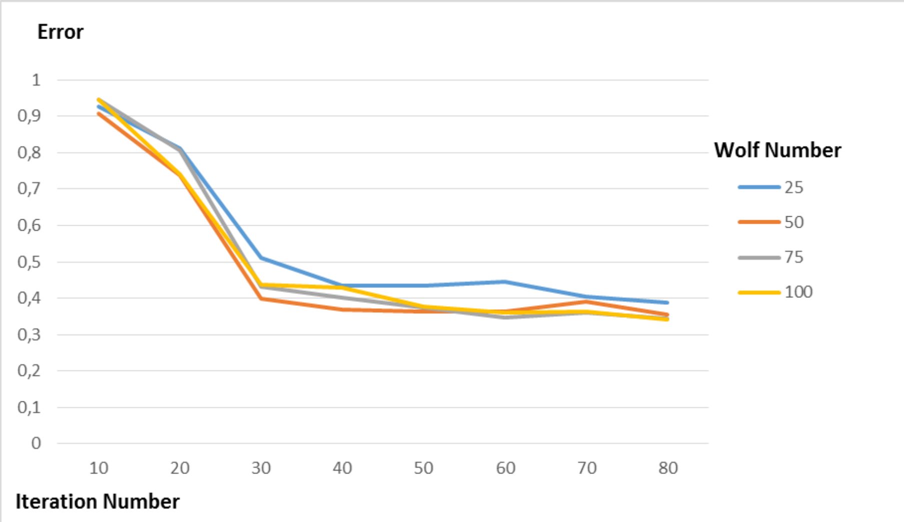

Figure 5.106. Results of GWO Color Quantizing n times iterations (Flowers)

Figure 5.107 represents the average of n*ten times iterations' results of GWO (Flowers) :

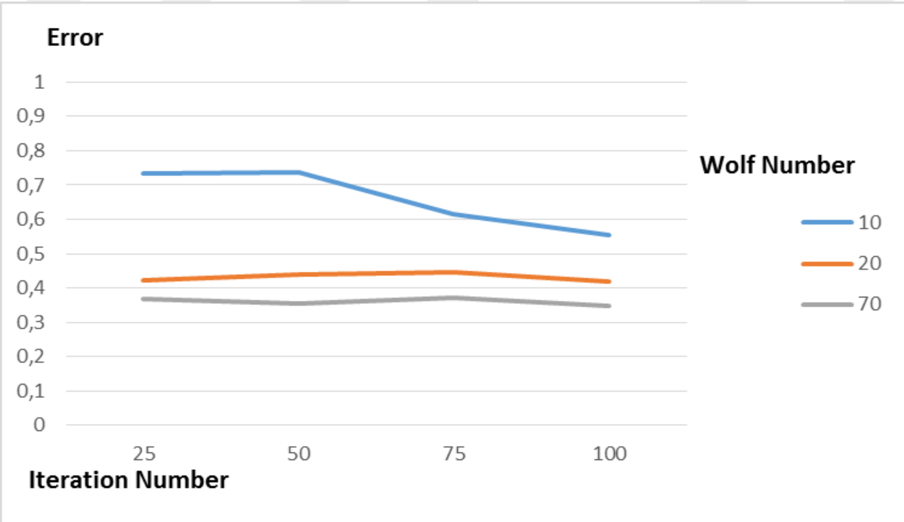

Figure 5.107. Average of n*ten times iterations' results of GWO (Flowers)

Figure 5.108 represents the median of n*ten times iterations' results of GWO (Flowers):

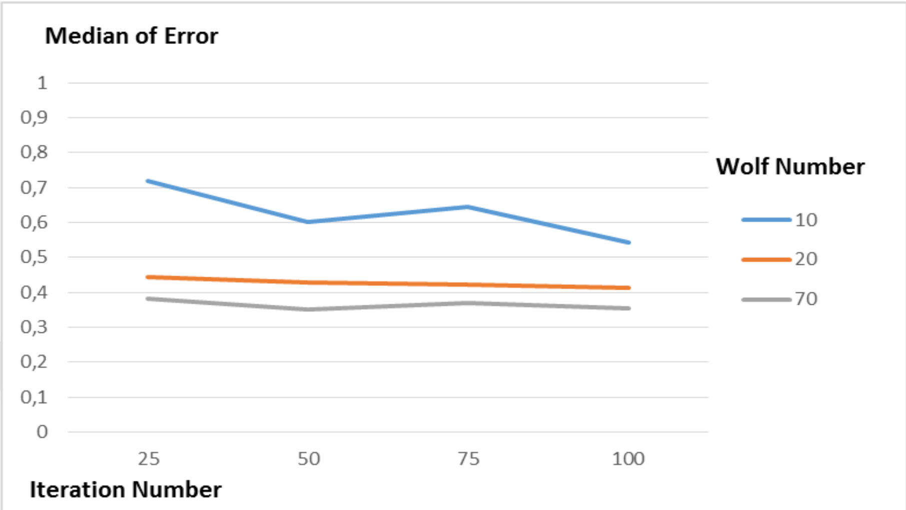

Figure 5.108**.** Median of n*ten times iterations' results of GWO (Flowers)

Figure 5.109 represents the STDDEV of n*ten times iterations' results GWO (Flowers)

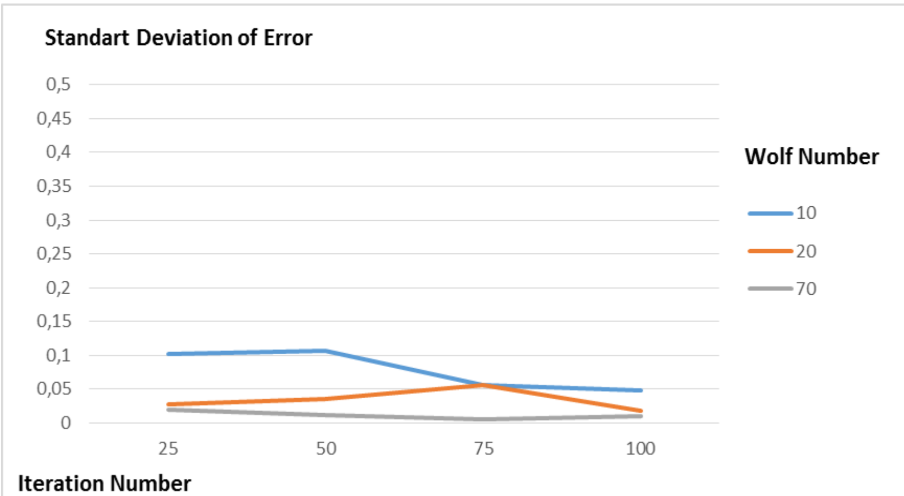

Figure 5.109. STDDEV of n*ten times iterations' results of GWO (Flowers)

Figure 5.110 represents the IOR of n*ten times iterations' results of GWO (Flowers)

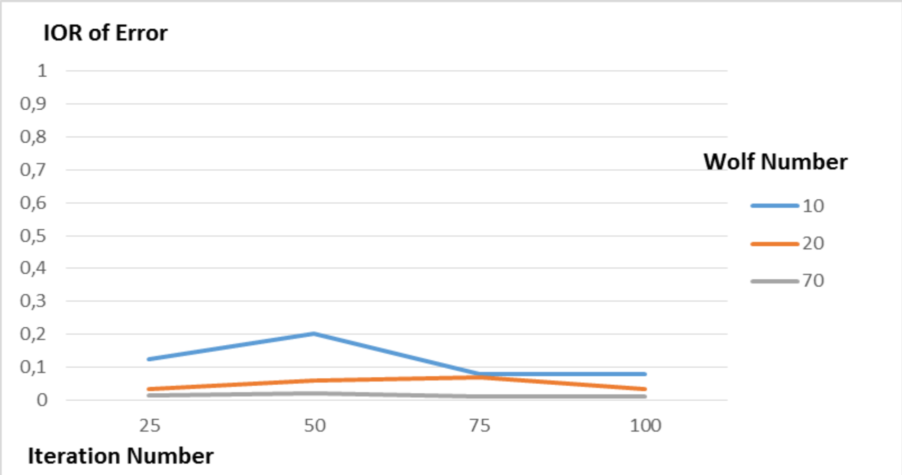

Figure 5.110. IOR of n*ten times iterations' results of GWO (Flowers)

Figure 5.111 represents Original Image Flowers

Figure 5. 111. Original Image Flowers

Figure 5.112 represents quantized image in GWO by 70 particals at 100 iteration

Figure 5. 112. Quantized image in GWO by 70 particals at 100 iteration

Figure 5.113 shows the results of GWO color quantizing n times iterations (Lenna):

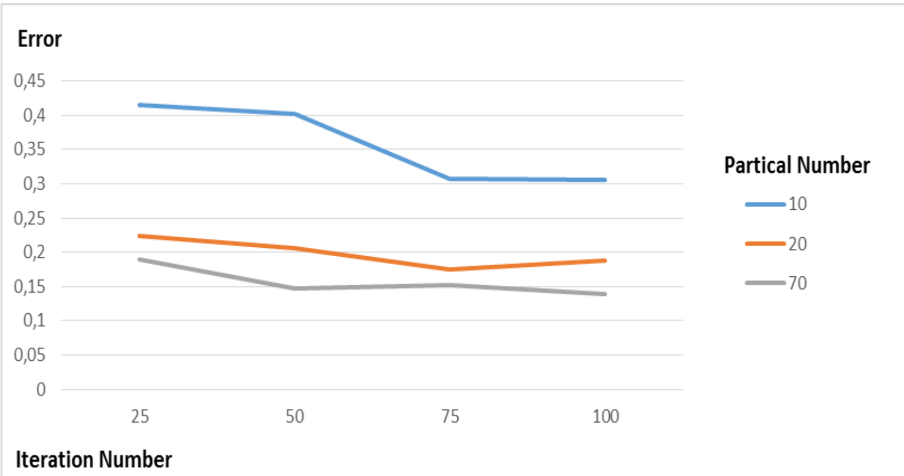

Figure 5.113. Results of GWO Color Quantizing n times iterations (Lenna)

Figure 5.114 represents the average of n*ten times iterations' results of GWO (Lenna):

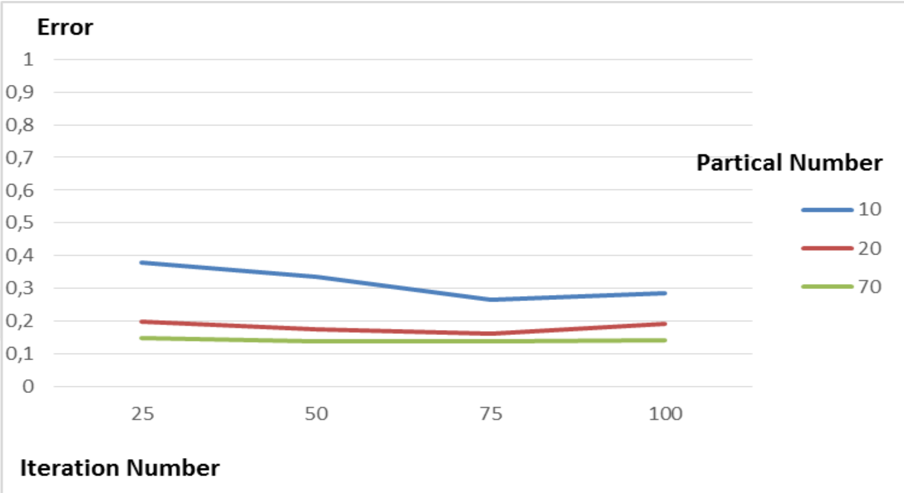

Figure 5.114. Average of n*ten times iterations' results of GWO (Lenna)

Figure 5.115 represents the median of n*ten times iterations' results of GWO (Lenna):

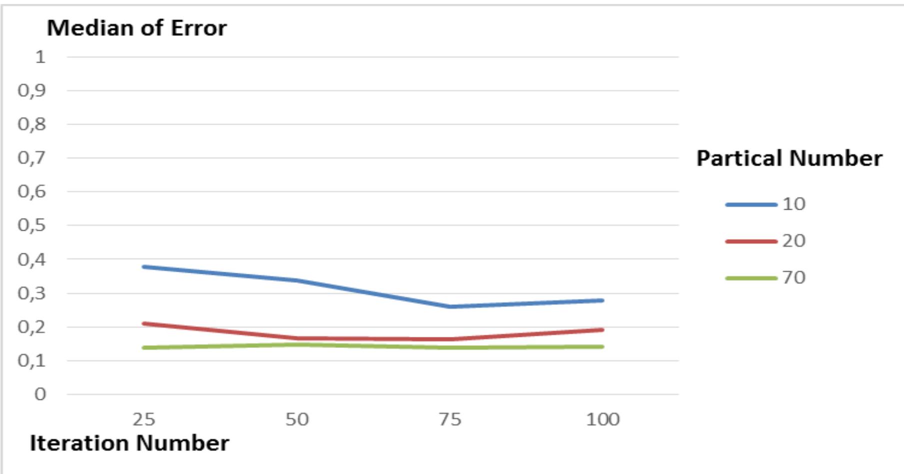

Figure 5.115.Median of n*ten times iterations' results of GWO (Lenna)

Figure 5.116 represents the STDDEV of n*ten times iterations' results GWO (Lenna):

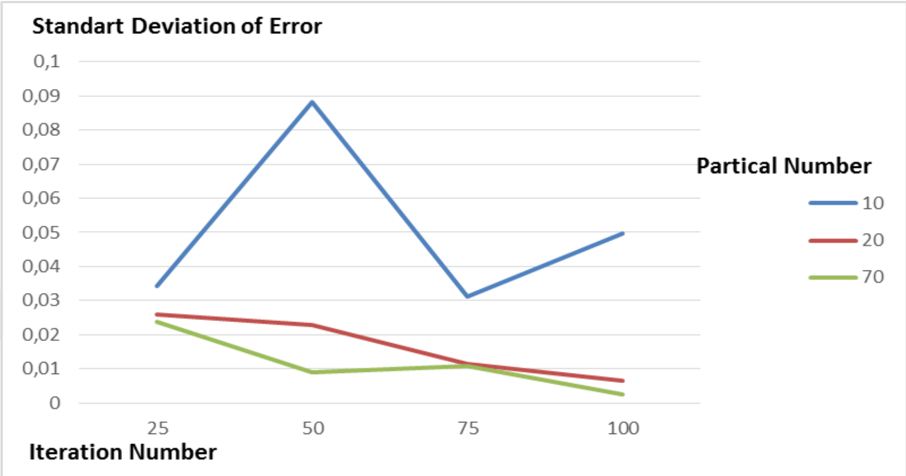

Figure 5.116. STDDEV of n*ten times iterations' results of GWO (Lenna)

Figure 5.117 represents the IOR of n*ten times iterations' results of GWO (Lenna):

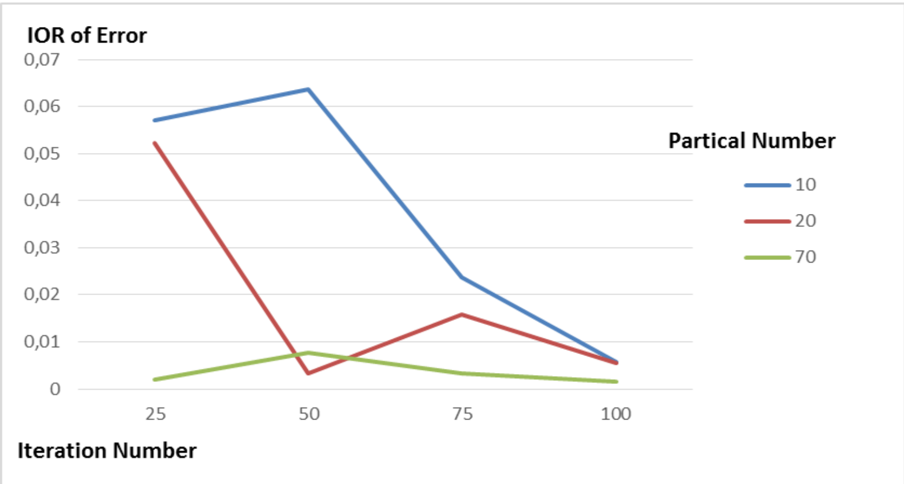

Figure 5.117. IOR of n*ten times iterations' results of GWO (Lenna)

Figure 5.118 Represents Original Image Lenna

Figure 5.118. Original Image Lenna

Figure 5.119 represents quantized image in GWO by 70 particals at 100 iteration

Figure 5.119. Quantized image in GWO by 70 particals at 100 iteration

Figure 5.120 shows the results of GWO color quantizing n times iterations (Baboon):

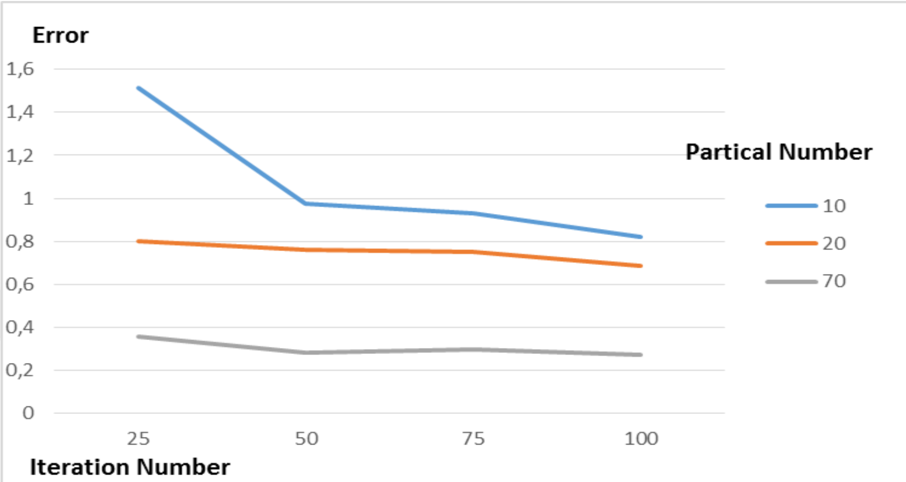

Figure 5. 120. Results of GWO Color Quantizing n times iterations (Baboon)

Figure 5.120 represents the average of n*ten times iterations' results of GWO (Baboon):

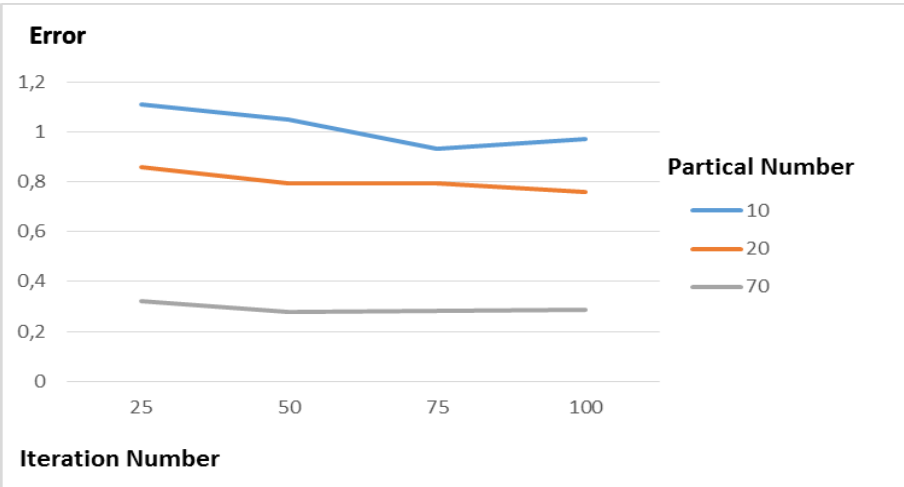

Figure 5.121Average of n*ten times iterations' results of GWO (Baboon)

Figure 5.122 represents the median of n*ten times iterations' results of GWO (Baboon):

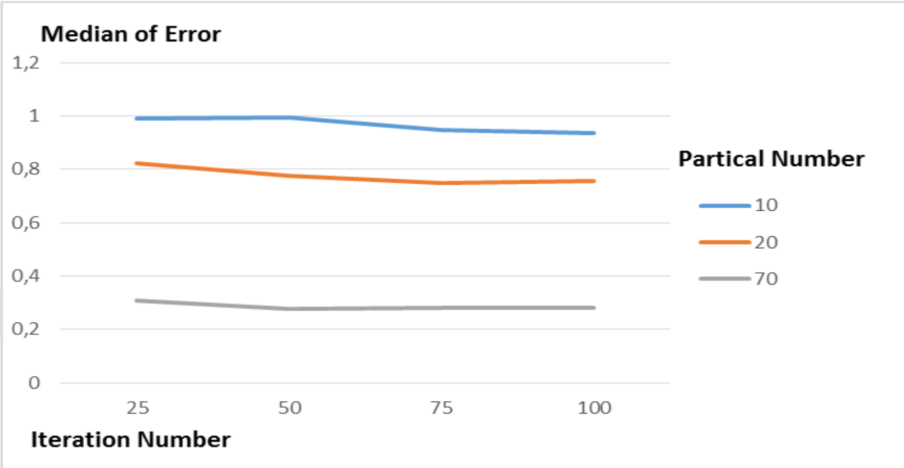

Figure 5. 122. Median of n*ten times iterations' results of GWO (Baboon)

Figure 5.123 represents the STDDEV of n*ten times iterations' results GWO (Baboon):

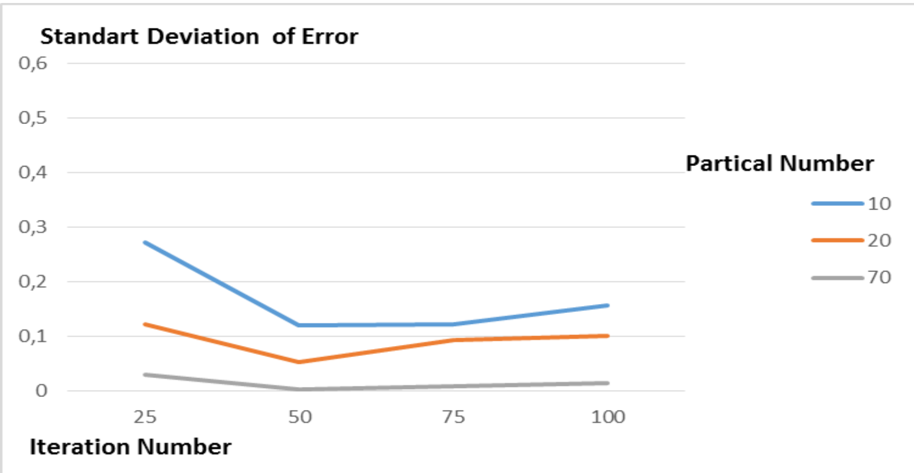

Figure 5.123**.** STDDEV of n*ten times iterations' results of GWO (Baboon)

Figure 5.124 represents the IOR of n*ten times iterations' results of GWO (Baboon):

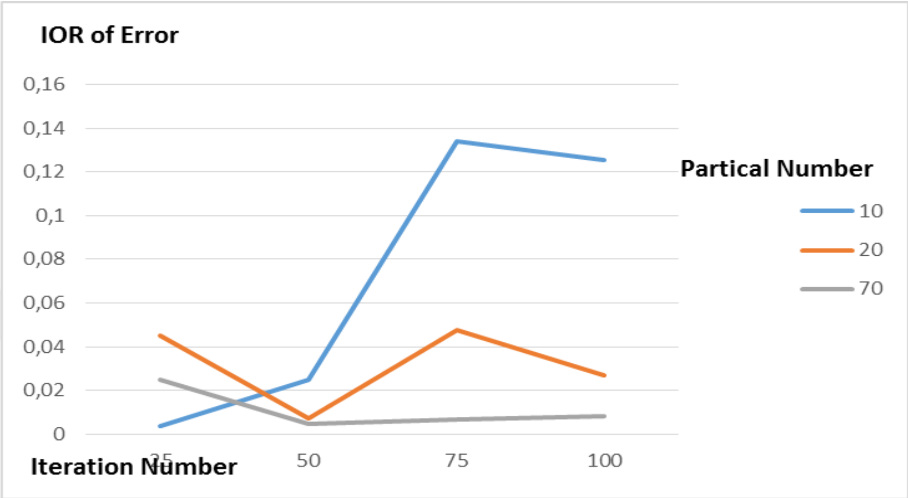

Figure 5.124. IOR of n*ten times iterations' results of GWO (Baboon)

Figure 5.125 represents Original Image Baboon

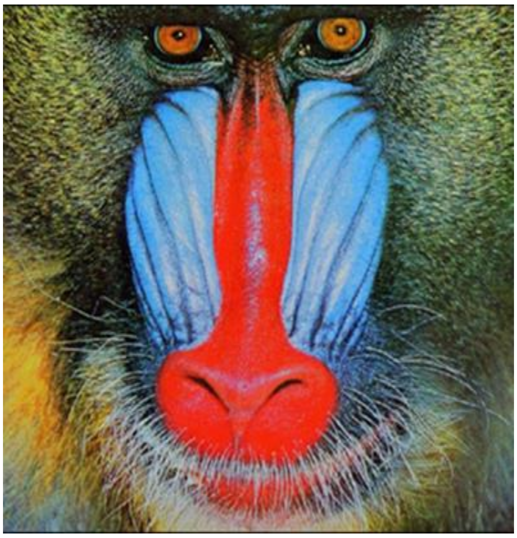

Figure 5.125. Original Image Baboon

Figure 5.126 represents quantized image in GWO by 70 particals at 100 iteration

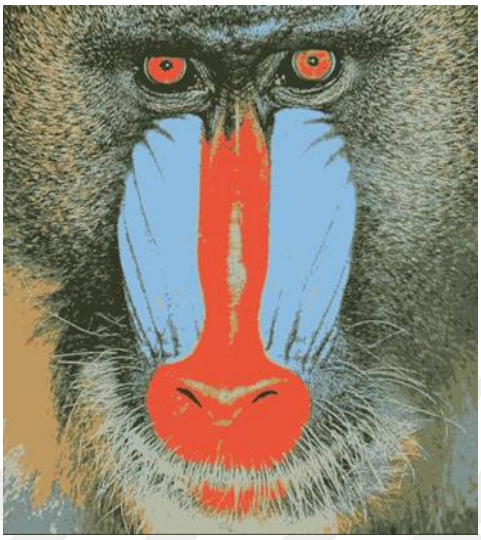

Figure 5.126. Quantized image in GWO by 70 particals at 100 iteration

6. CONCLUSION

In this thes is color quantization process is implemented by three meta-heuristic algorithms. Each steps of algorithm are applied to achiving best palette for quantizing.

Researches showed that the best quantized image had reached by Grey Wolf Algorithm with 0.34 error by 70 wolves in 100^{th} iteration for Flowers image,0.14 error by 70 wolves in 100^{th} iteration for Lenna image,0.28 error by 70 wolves in 100^{th} iteration for Baboon image.

Second best quantized image had reached by Partical Swarm Optimization Algorithm with 0.84 error by 20 particals in 100^{th} iteration for Flowers image,0.35 error for Lenna image by 20 particals in 50^{th} iteration,0.62 error for Baboon image by 20 particals in 100^{th} iteration.

Last best quantized image had reached by Differential Evolution Algorithm with 1.33 error by 30 partical at CR: 0.4 in 50^{th} iteration in Best_2_Bin mutation strategy for Flowers image.

Lenna image error at CR: 0.2 at 100^{th} iteration at Best_2_Bin mutation strategy.

Baboon image error ar CR: 0.1 at 25^{th} iteration at Best_1_Bin mutation stategy.

In mentioned above algorithms are ordered by the error value of best palettes. According to the time-consuming criteria Differential Evolution Algorithm had been the best algorithm and then Grey Wolf Algorithm is the second best and Partical Swarm Algorithm had been the last algorithm. Because of time-consuming of algorithms, PSO could be implemented at just 20 particals, GWO could be implemented at 80 number wolves. At the same number of particals in PSO had reached less error with same number of wolves in GWO.

At the end of the survey GWO had reached the best performance in quantizing. Time-consuming problem also had been better than PSO. Because of the strict priorty of wolves, it had less standard deviation, less difference between interquartile ranges of GWO at randomly initialization step. This could show GWO had more stable results than the other meta-heuristic algorithms. Finally, we may express that GWO algoritm had been the best resulted between these three of meta-heuristic algorithms.

In this work in order to avoid time consuming in meta-heuristic algorithm clustering methods are had not been applied. In order to have much better colormaps in GWO, by post or pre clustering new colormaps could be selected to quantize image with less error. But time consuming could be about to stay however clustering methods.

REFERENCES

[1] M. Yasin ÖZSAĞLAM, Mehmet ÇUNKAŞ , 2008, Optimizasyon Problemlerinin Çözümü için Parçaçık Sürü Optimizasyonu Algoritması , Cilt:11 Sayı: 4 s.299-305

[2] Qinghua Su and Zhongbo Hu **,** 2013, Color Image Quantization Algorithm Based on Self-Adaptive Differential Evolution , Volume 2013, Article ID 231916

[3] Özgür BAŞKAN, Hüseyin CEYLAN ,2014, Differential Evolution Algorithm Based Solution Approaches For Solving Transportation Network Design Problems, Cilt:20 (9) Sayı: 4 s.324-331

[4] Seyedali Mirjalili, Seyed Mohammad Mirjalili, Andrew Lewis, 2014 ,Grey Wolf Optimizer , vol. 69, pp. 46-61, 2014.

[5] https://www.muhendisbeyinler.net/parcacik-suru-optimizasyonu , March 2016

[6] http://lmcaraig.com/color-quantization-using-k-means, March 2016

[7] Muro C, Escobedo R, Spector L, Coppinger R. Wolf-pack (Canis lupus) hunting strategies emerge from simple rules in computational simulations. Behav Process 2011,88:192–7.

[8] Wu X, Zhang K (1991) A better tree-structured vector quantizer, Proceedings IEEE Data Compression Conference, 392-401.

[9] Shafer S, Kanade T (1987) Color Vision, Encyclopedia of Artificial Intelligence, Wiley.

[10] Celenk M (1990) A color clustering technique for image segmentation, computer vision, Graphics and Image Processing 52: 145-170.

[11] Freisleben B, Schrader A (1997) An evolutionary approach to color image quantization, Proceedings of IEEE International Conference on Evolutionary Computation, 459-464.

[12] Xiang Z, Joy G (1994) Color image quantization by agglomerative clustering, IEEE Computer Graphics and Applications 14(3): 44-48

[13] R. Storn andK. Price, “Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces,” Journal of Global Optimization, vol. 11, no. 4, pp. 341–359, 1997.

[14] Y. Shen, M. Li, and H. Yin, “A novel differential evolution for numerical optimization,” International Journal of Advancements in Computing Technology, vol. 4, no. 4, pp. 24–31, 2012.

[15] H. Li and J. Tang, “Improved differential evolution with local search,” Journal of Convergence Information Technology, vol. 7, no. 4, pp. 197–204, 2012.

[16] M. G. H. Omran, A. P. Engelbrecht, andA. Salman, “Differential evolution methods for unsupervised image classification,” in Proceedings of the IEEE Congress on Evolutionary Computation (IEEE CEC ’05), pp. 966–973, September 2005.

[17] S. Das and A. Konar, “Automatic image pixel clustering with an improved differential evolution,” Applied Soft Computing Journal, vol. 9, no. 1, pp. 226–236, 2009. 8 Computational Intelligence and Neuroscience

[18] Q. Su, Z. Huang, and Z. Hu, “Binarization algorithm based on differential evolution algorithm for gray images,” ICNC-FSKD, vol. 6, pp. 2624–2628, 2012.

[19] J. Brest, S. B. Boˇskovi´c, V. ˇZume, and M. S. Mauˇcec, “Performance comparison of self-adaptive and adaptive differential evolution algorithms,” Soft Computing, vol. 11,no. 7, pp. 617–629, 2007.

[20] J. Brest, V.ˇZumer, andM. S. Mauˇcec, “Self-adaptive differential evolution algorithm in constrained real-parameter optimization,” in Proceedings of the IEEE Congress on Evolutionary Computation (CEC ’06), pp. 215–222, July 2006.

[21] S.-C. Cheng and C.-K. Yang, “A fast and novel technique for color quantization using reduction of color space dimensionality,” Pattern Recognition Letters, vol. 22,no. 8, pp. 845–856, 2001.

CURRICULUM VITAE

I was born in İstanbul in 1987.I graduated from the departman of Computer Engineering of Near East University in 2008.I had started to study master program in Gebze Yüksek Teknoloji University in 2010.In 2015 I passed by master program Çukurova University in 2015.

I have been worked in KALE Holding A.Ş. in 2009-2013 as Software Developer.I returned my company at 2015 as Senior Software Developer.I am still working at KALE Holding A.Ş.