وزارة التعليم العالي والبحث العلمي

جامعة ديالى

كلية التربية للعلوم الصرفة

قسم علوم الحاسوب

Edge Detection Using Bat Algorithm

بحث مقدم الى قسم الحاسوب وهو جزء من متطلبات نيل

شهادة البكالوريوس في علوم الحاسوب

من قبل الطالبان

أحمد علي صالح عباس فاضل رشيد

بإشراف

م.د عادل عبد الغني

١٤٣٩هـ ٢٠١٨م

2.1 Introduction

Image is formed in the eye and in the camera by the amount of illumination reflected by an object. In computer vision, image processing is any form of signal processing for which the input is an image, such as photographs or frames of videos. The output of image processing can be either an image or a set of characteristics or parameters related to image. The image processing techniques like image restoration, image enhancement, image segmentation edge detection …ect [10].

The edge detection is widely applied in areas like image recognition, image classification, image enhancement, and in pattern recognition in general. Applying gradient operators on images can result in image edges when the edge gradient values exceed some defined thresholds.

2.2 Edge Models

Large changes in image brightness over a short spatial distance indicates the presence of an edge. Edge is a set of connected pixels that lay on the boundary between two regions whose gray level has outstanding change. The edge can by located between objects and background or two objects. Edge models are classified according to their intensity profiles. A step edge involves a transition between two intensity levels occurring ideally over the distant of 1 pixel.

Figure 2-1(a) shows a section of a vertical step and a horizontal intensity profile though the edge. Step edges occur, for example, in images generated by a computer for use in areas such as solid modelling and animation. These clean, ideal edges can occur over the distance of 1 pixel, provided that no additional processing (such as smoothing) is used to make them look real. Digital step edges are used frequently as edge models in algorithm development.

In practice, digital images have edges that are blurred and noisy, determined by limitations in the focusing mechanism and noise level determined by the electronic components of imaging system. In such situations, edges are more closely modeled as having an intensity ramp profile such as edge in Figure 2-1(b). The slope of the ramp is inversely proportional to the degree of blurring in the edge. It can be seen that in this model, there is no thin (1 pixel thick) path. Instead, an edge point which is now any point contained in the ramp, and an edge segment would then be a set of such points that are connected. A third model of an edge is called roof edge, having the characteristics illustrated in Figure 2-1(c). Roof edges are models of lines through a region, with the base(width) of a roof edge being determined by the thickness and sharpness of the line .The figures below show the various edge models [10,11].

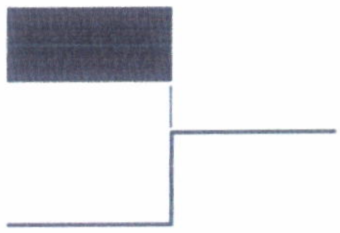


Figure 2-1(a): Step Edge with intensity profile

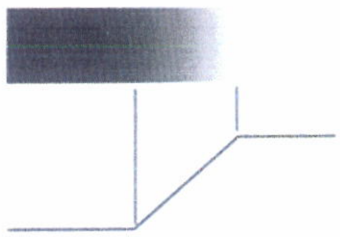


Figure 2-1(b): Ramp Edge with intensity profile

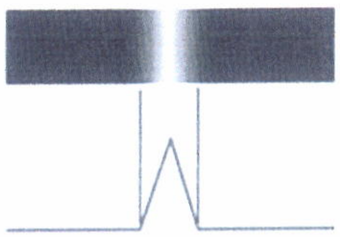


Figure 2-1(c): Roof Edge with intensity profile

2.3 Edge Detection

Edge detection is considered as a paramount step in many image transforming functions. The information obtained from it given as an input to feature extraction and object segmentation for further processing [12]. For edge characterization, the intensity function is calculated and analyzed to identify the sharp discontinuities in an image. For better results, the considered edges should be of good quality as much as possible. When an image is reduced to its edge pixels, it saves memory, as well as these edge details, can be used in feature detection and object recognition. Edge pixels, which play an important role in image analysis, are determined by practicing boundary specification of Sobel , Prewitt , Kirsch and Canny .. etc. These rules certainly benefit the purpose but increased the significant overhead of eliciting and managing massive data as compared to the simpler fuzzy methods [13].

Swarm-based algorithms are recent, nature-influenced, culture-based algorithms. Swarm Intelligence (SI) is used to illustrate the course of communal swarms in nature, such as ant colonies, bats, and bird flocks. Swarm Intelligence principles have applications in various problem domains, for example, the discovery of most favorable path, function optimization problems, anatomical commutation, image and data dissection, and scheduling [5]. Computational modeling of swarms has been gaining popularity in a broad-amplitude of distinct realms, for example- in bioinformatics domain, in medical informatics, in dynamical systems, and operations research.

In the recent times, various swarm intelligence algorithms are designed and are used for distinct purposes the popular ones are- Ant Colony Optimization (ACO), Gravitational Search Algorithm (GSA) and Bat Algorithm (BA).

As stated earlier, Bat algorithm is energized by the trait of echolocation attitude of bats. This action helps them to operate and collect their necessities even in the absence of light The echolocation of the bats is observed as:

Every implicit bat shoots with an acceleration pi at location (solution) zi with a varying recurrence or wavelength and vibration Vi. As it hunts and discovers its food, it changes frequency, loudness and pulse emission rate ri. The bats take a local random walk intensifying the search. Until the convinced break stone is faced, selection of the best continues. The frequency-tuning approach is used to curb the disciplined progressive conduct of throng of bats.

Bat Algorithm uses a combination of frequency tuning technique and automatic zooming that empowers it to elaborate distinct solution in the community and provides symmetry in analysis and exploitation in the interim of the hunt operation by imitating or simulating the disparity of pulse emission rates and loudness of bats when hunting for prey. As an outcome, it confirms to be very competent and powerful with an exemplary abrupt outset [4].

In this research, Bat Algorithm is used to resolve the edge detection problem in an optimal way. The local asymmetry in image acuteness caused by the movement of agents in search of food and prey is used to reveal the edge pixels [8].

(أ) في الخوارزمية الجديدة pixel واحد يمثل خفاش الخوارزمية القديمة الصف بكامله خفافيش

(ب) في الجديدة نعوض pixel في ثلاث معادلات
القديمة صف كامل في ثلاث معادلات

الخوارزمية القديمة

2.4 The Standard Bat Algorithm

The standard bat algorithm, developed by Xin-She Yang in 2010, was based on the echolocation or bio-sonar characteristics of microbats . Before we outline the details of the algorithm, let us briefly introduce the echolocation.

2.4.1 Echolocation of Microbats

Most bats uses echolocation to a certain degree; among all the species, microbats use echolocation extensively, while megabats do not.

Microbats typically use a type of sonar, called, echolocation, to detect prey, avoid obstacles, and locate their roosting crevices in the dark. They can emit a very loud sound pulse and listen for the echo that bounces back from the surrounding objects (Richardson, 2008).Their pulses vary in properties and can be correlated with their hunting strategies, depending on the species. Most bats use short, frequency-modulated signals to sweep through about an octave, and each pulse lasts a few thousandths of a second (up to about 8 to 10 ms) in the frequency range of 25 kHz to 150 kHz. Typically, microbats can emit about 10 to 20 such sound bursts every second, and the rate of pulse emission can be sped up to about 200 pulses per second when homing on their prey. Since the speed of sound in air is about $\upsilon = 340$ m/s, the wavelength λ of the ultrasonic sound bursts with a constant frequency f is given by $\lambda = \upsilon / f$, which is in the range of 2mm to 14mm for the typical frequency range from 25 kHz to 150 kHz. Interestingly, these wavelengths are in the same order of their prey sizes.

Though in reality microbats can also use time delay between their ears and loudness variations to sense three-dimensional surroundings, we are mainly interested in some features of echolocation so that we can some link them with the objective function of an optimization problem, which makes it possible to formulate a smart, bat algorithm[4].

2.4.2 Bat Algorithm

Based on the above description and characteristics of bat echolocation, Xin-She Yang (2010) developed the bat algorithm with the following three idealized rules:

1. All bats use echolocation to sense distance, and they also `know' the difference between food/prey and background barriers in some magical way.
2. Bats fly randomly with velocity υ_i at position x_i with a frequency f (or wavelength λ) and loudness A_o to search for prey. They can automatically adjust the wavelength (or frequency) of their emitted pulses and adjust the rate of pulse emission $r \in [0 ; 1]$, depending on the proximity of their target.
3. Although the loudness can vary in many ways , we assume that the loudness varies from a large (positive) A_o to a minimum constant value A_{min}.

For simplicity, we do not use ray tracing in this algorithm, though it can form an interesting feature for further extension. In general, ray tracing can be computationally extensive, but it can be a very useful feature for computational geometry and other applications. Furthermore, a given frequency is intrinsically linked to a wavelength. For example, a frequency range of [20kHz , 500kHz] corresponds to a range of wavelengths from 0.7mm to 17mm in the air. Therefore, we can describe the changes either in terms of frequency f or wavelength λ to suit different applications, depending on the ease of implementation and other factors [4].

2.4.3 Bat Motion

Each bat is associated with a velocity υ_i^t and a location x_i^t , at iteration t , in a d-dimensional search or solution space. Among all the bats, there exists a current best solution x*. Therefore, the above three rules can be translated into the updating equations for x_i^t and velocities υ_i^t :

$$f_i = f_{min} + (f_{max} - f_{min})\,\beta \quad \text{--------} \quad (1)$$

$$\upsilon_i^t = \upsilon_i^{t-1} + (X_i^{t-1} - X^*)\,f_i \quad \text{---------} \quad (2)$$

$$X_i^t = X_i^{t-1} + \upsilon_i^t \quad \text{-------------------} \quad (3)$$

Where $\beta \in [0,1]$ is a random vector drawn from a uniform distribution.

As mentioned earlier, we can either use wavelengths or frequencies for implementation, we will use fmin = 0 and fmax = 0 (1), depending on the domain size of the problem of interest. Initially, each bat is randomly assigned a frequency which is drawn uniformly from [fmin , fmax]. For this reason, bat algorithm can be considered as a frequency-tuning algorithm to provide a balanced combination of exploration and exploitation. The loudness and pulse emission rates essentially provide a mechanism for automatic control and auto-zooming into the region with promising solutions[4,9,14].

2.4.4 Local Search of bat

A random walk with direct exploitation is used for the local search that modifies the current best solution according the equation:

$$X_{new} = X_{old} + \partial A^t \quad \text{--------------} \quad (4)$$

where $\partial \in [-1,1]$ is a random number, while A^t is the average loudness of all the best at this time step[4,9,14].

2.4.5 Variations of Loudness and Pulse Rates

In order to provide an effective mechanism to control the exploration and exploitation and switch to exploitation stage when necessary, we have to vary the loudness A_i and the rate r_i of pulse emission during the iterations.
Since the loudness usually decreases once a bat has found its prey, while the rate of pulse emission increases, the loudness can be chosen as any value of convenience, between A_{min} and A_{max}, assuming $A_{min} = 0$ means that a bat has just found the prey and temporarily stopped emitting any sound.
With these assumptions, we have

$$A_i^{t+1} = \alpha A_i^t \quad \text{---------------------} \quad (5)$$

and

$$r_i^t = r_i^o [1 - \exp(-\Upsilon t)] \quad \text{--------} \quad (6)$$

where α and Υ are constants.

A_i^t and A_i^{t+1} are the loudness at the proceed and present iteration, respectively.

r_i^o is the initial pulse emission rate.

r_i^t is the pulse emission rate at the iteration t.

In essence, here α is similar to the cooling factor of a cooling schedule in simulated annealing. For any $0 < \alpha < 1$ and $\Upsilon > 0$[4,9,14,15]

.5 General bat algorithm

```
Objective function $f(X)$ , $X = (X_1, \dots, X_d)^t$
Initialize the bat population $X_i$ ($i = 1, 2, \dots, n$) and $V_i$
Define pulse frequency $f_i$ at $X_i$
Initialize pulse rates $r_i$ and the loudness $A_i$
While ($t <$ Max number of iterations )
Generate new solutions by adjusting frequency,
and updating velocities and locations / solutions [equations (2) to (4) ]
    if ($rand > r_i$)
    Select a solution among the best solutions
    Generate a local solution around the selected best solution
    end if
    Generate a new solution by flying randomly
    if ($rand < A_i$ & $f(X_i) < f(X_*)$)
    Accept the new solutions
    Increase $r_i$ and reduce $A_i$
   end if
Rank the bats and find the current best $X_*$
end while
Postprocess results and visualization
```

الاشياء التي استخدمناها

3.1 Proposed approach

1) Given an image of (x*y) pixels.

2) Taken n of pixels as a population that is distributed over all image.Each one of these pixels represents a bat having two values X and Y.

3) The values of X and Y are the X-axis and Y-axis positions of the bat (pixel).

4) For each bat, suppose the following values :-

The pulse rate ri ,velocity Vi , loudness Ai, minimum frequency (f_{min}) and maximum frequency (f_{max}).

5) calculate values of ri , Ai , fi for each bat as follows:-

A) ri represents the largest difference in intensity between the bat (pixel) and it's 8 neighbors.

B) Ai represents the smallest difference in intensity between the bat (pixel) and it's 8 neighbors.

C) fi from equation:-

$$F_i = f_{min} + (f_{max} - f_{min})\,\beta$$

Where β is random value.

6) Determine is the value of the best pixel for each bat, which initially represent the value of the bat itself and this will be updated when the bat moves to new pixels in the search for the goal. Where it will always represents the best pixel passed by the bat . this priority is measured base on the pulse rate (from step 5).

7) Start a main loop with total number of repeats tmax , each repetition includes n of bats (pixels).

8) Start a loop represents number of bats $i=1 \rightarrow i=n$, n is the number of bats for the population.

9) Calculate a new value for the bat (pixel) using equations 1, 2 and 3 .

10) Calculate the new pulse rate for the resulting point of step 9 .

11) Compare the best pulse rate with the pulse rate of the new pixel to decide whether reject or accept the new pixel.

If it is rejected, go to local search in equation 4 to find anew pixel.

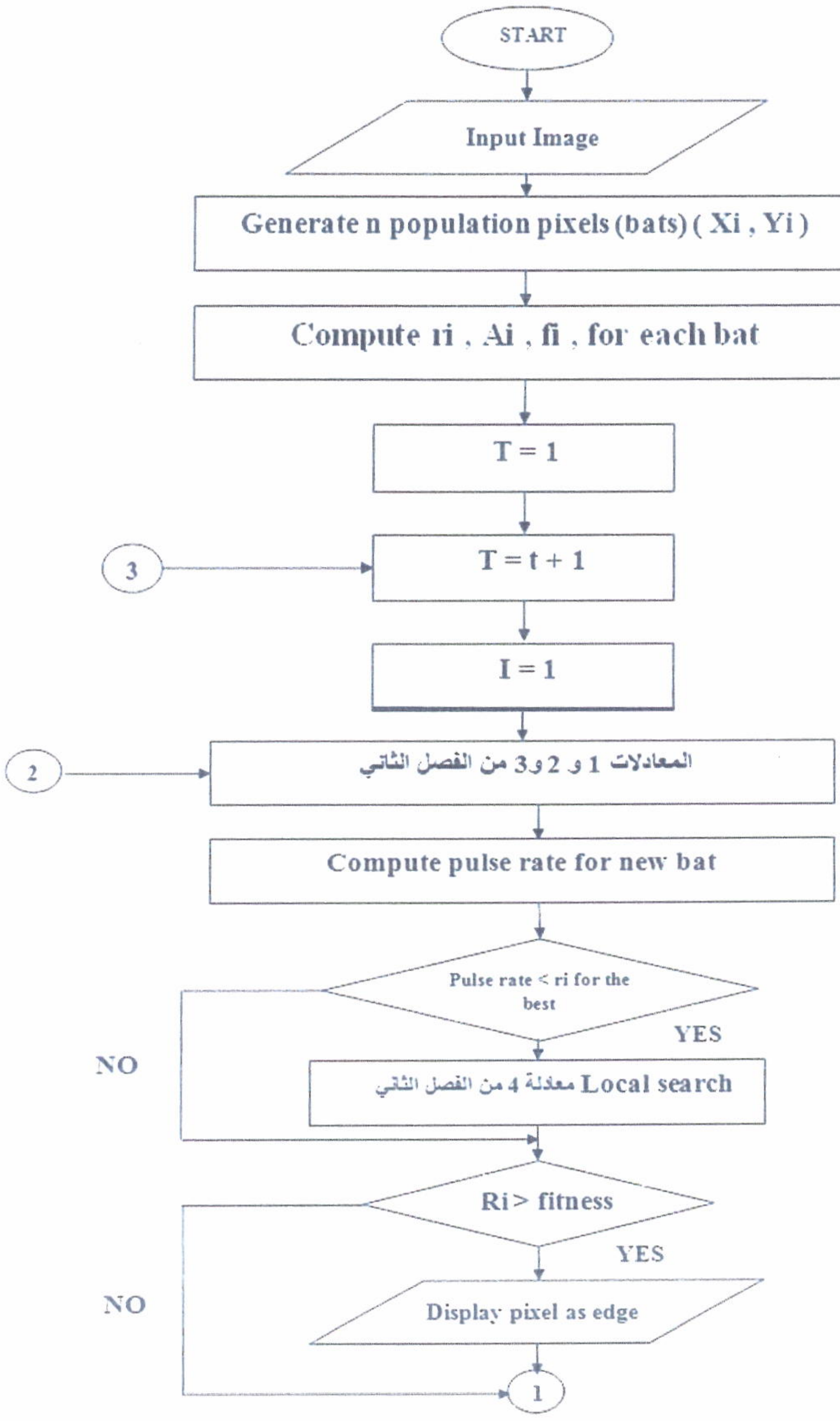
12) Check if the new pixel obtained from step 11 to be an edge or not by comparing its gray level with the fitness that depends on a given threshold value.

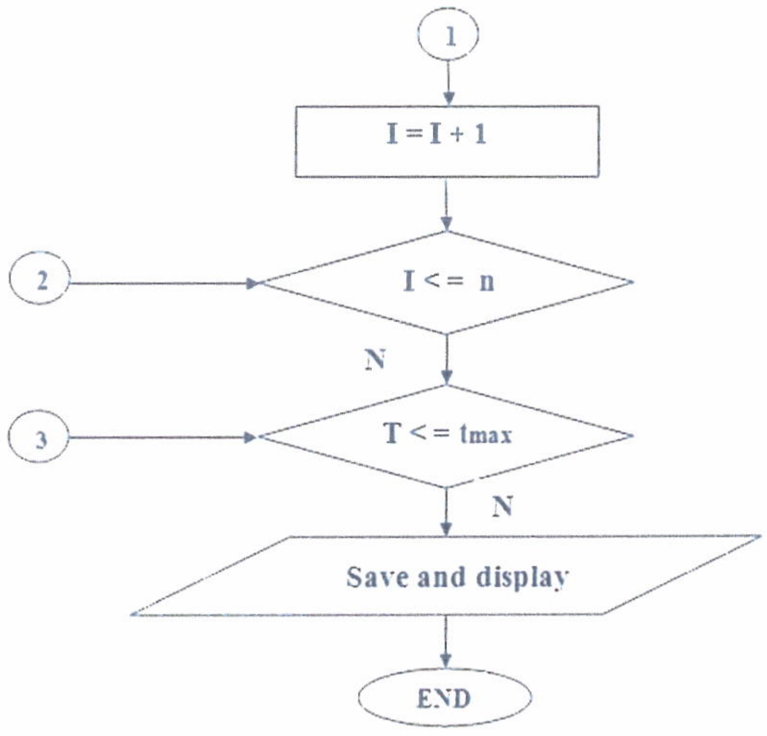
13) Repeat steps (9-11) for each bat and for n bats.

14) Repeat steps (8-13) to the value of t max.

15) Arrange the display of results that represent the edges of the image.

3.2 Flowchart of proposed approach


START
Input Image
Generate n population pixels (bats) (Xi , Yi)
Compute ri , Ai , fi , for each bat
T = 1
3
T = t + 1
I = 1
2
المعادلات 1 و 2 و3 من الفصل الثاني
Compute pulse rate for new bat
Pulse rate < ri for the best
NO
YES
معادلة 4 من الفصل الثاني Local search
Ri> fitness
NO
YES
Display pixel as edge
1



3.3 Experimental results :

Bat algorithm was implemented in visual basic programming language . Different images are taken as the test images . Parameters of the Bat algorithm are population size (n) which represents the number of bats (pixel of image) , the maximum number of iterations tmax , loudness (A) , pulse emission rate (r), minimum frequency fmin , maximum frequency f max and velocity (v). We used a random size of population and number of iteration (t max = 100) , fmin=0 , fmax=2, velocity=1.

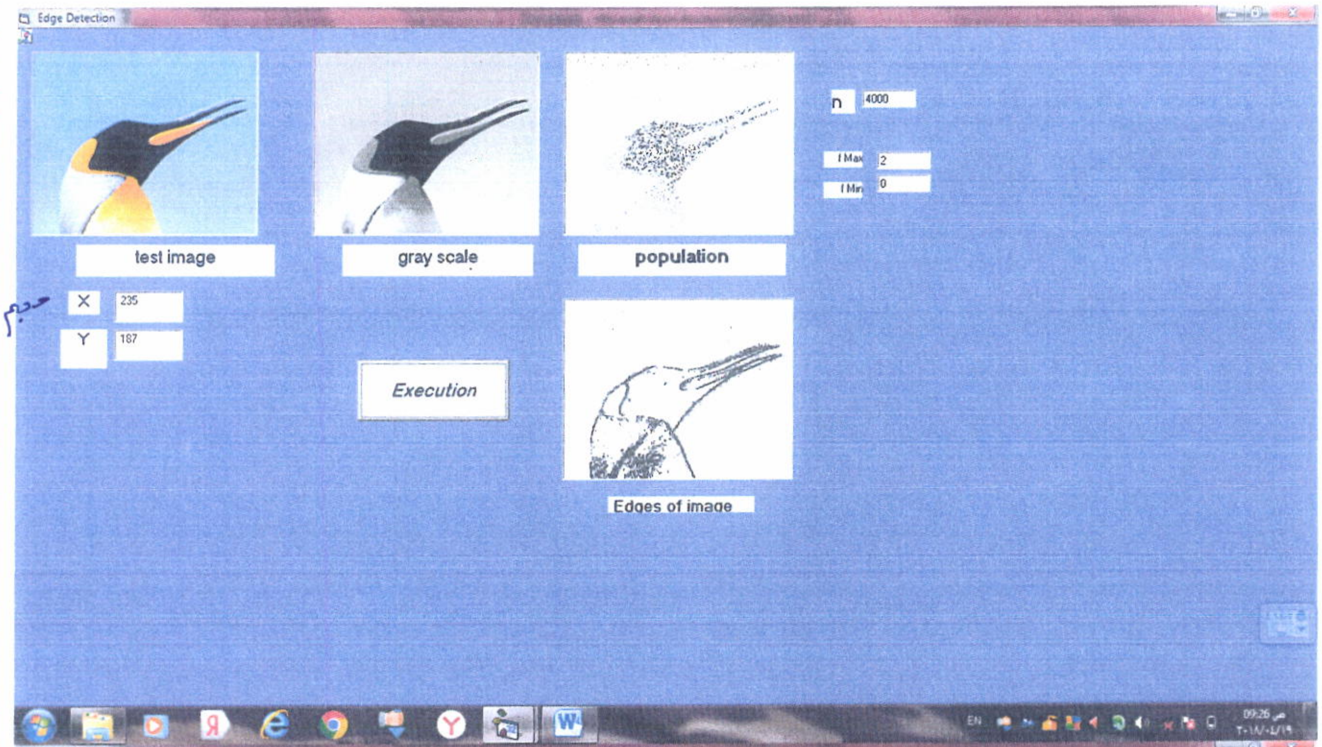
Implementation of Bat algorithm code obtained the edges of test images shown in following figures .

3.4Algorithm of bat

1) given an image of (x*y) pixels.
2) taking n of pixels as a population.
That is distributed over all image. each one of these pixels represents a bat having
two values X and Y.
3) the values of X and Y are the X-axis and Y-axis positions of the bat (pixel).
4) for each bat, suppose the following values :-
The pulse rate ri ,velocity Vi , loudness Ai, and minimum frequency and maximum frequency fmin and fmax.
5) calculate values of ri , Ai , fi for each bat as follows:-
A) ri represents the largest difference in intensity between the bat (pixel) and it's 8 neighbors.
B) Ai represents the smallest difference in intensity between the bat (pixel) and it's 8 neighbors.
C) fi from equation:-
$fi=fmin +(fmax - fmin) \beta$
Where β is random value.
6) Determine is the value of the best pixel for each bat, which initially represent the value of the bat itself and this will be updated when the bat moves to new pixels in the search for the goal. Where it will always represents the best pixel passed by the bat . this priority is measured base on the pulse rate (from step 5).
7) Start a main loop with total number of repeats tmax , each repetition includes n of bats (pixels).
8) Start a loop represents number of bats $i=1 \rightarrow i=n$, n is the number of bats for the population.
9) Calculate a new value for the bat (pixel) using equations 1& 2 & 3
10) Calculate the new pulse rate for the resulting point of step 9 .
11) Compare the best pulse rate with the pulse rate of the new pixel and therefore reject or accept the new pixel
if it is rejected, go to local search in equation 4 to find a new pixel.
12) Check if the new point from step 11 of the bat is an edge or not by comparing it with the fitness that depends on a given threshold value.
13) Repeat steps (9-11) for each bat & for n of bats.
14) Repeat steps (8-13) to the value of t max.
15) Arrange the display of results that represent the edges of the image.

طريقتنا الخوارزمية على عدة صور

The program was applied several different images in their characteristics and the results were as in below


Edge Detection

test image | gray scale | population

حجم

X 235

Y 187

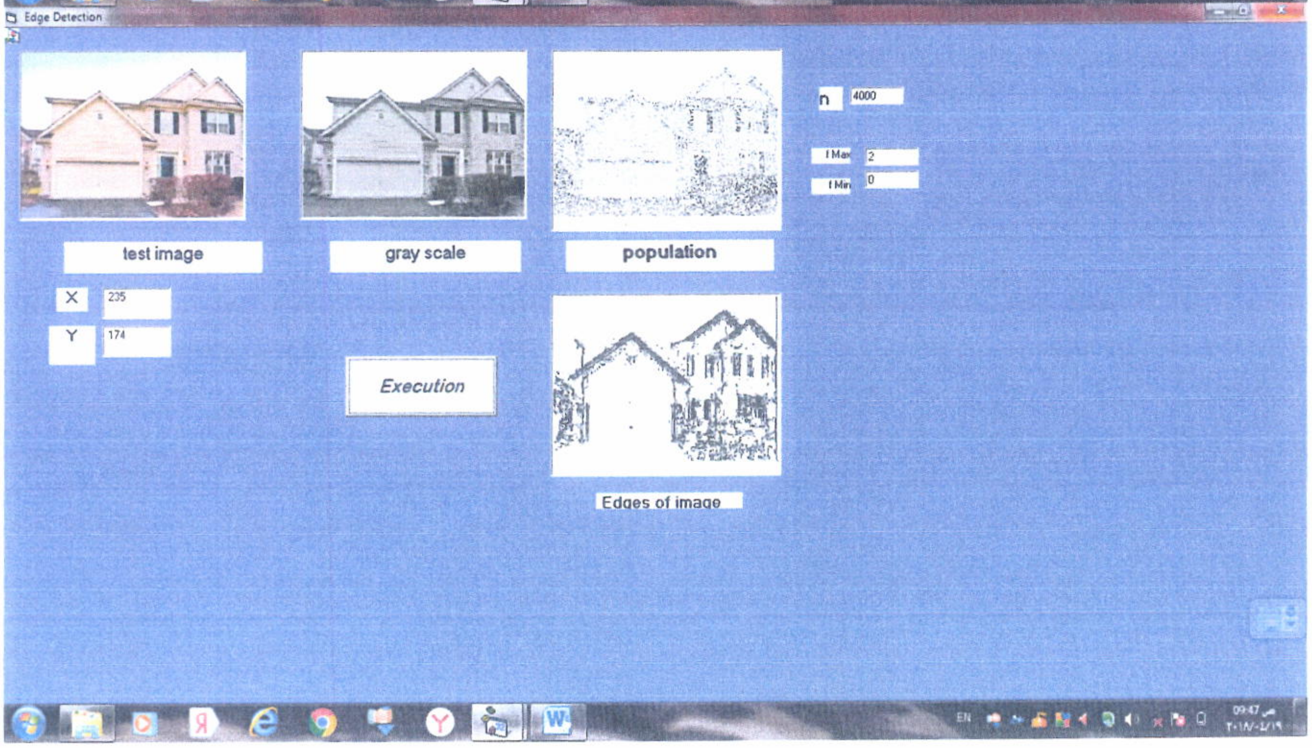
n 4000

I Max 2

I Min 0

Execution

Edges of image


Edge Detection

test image | gray scale | population

X 235

Y 174

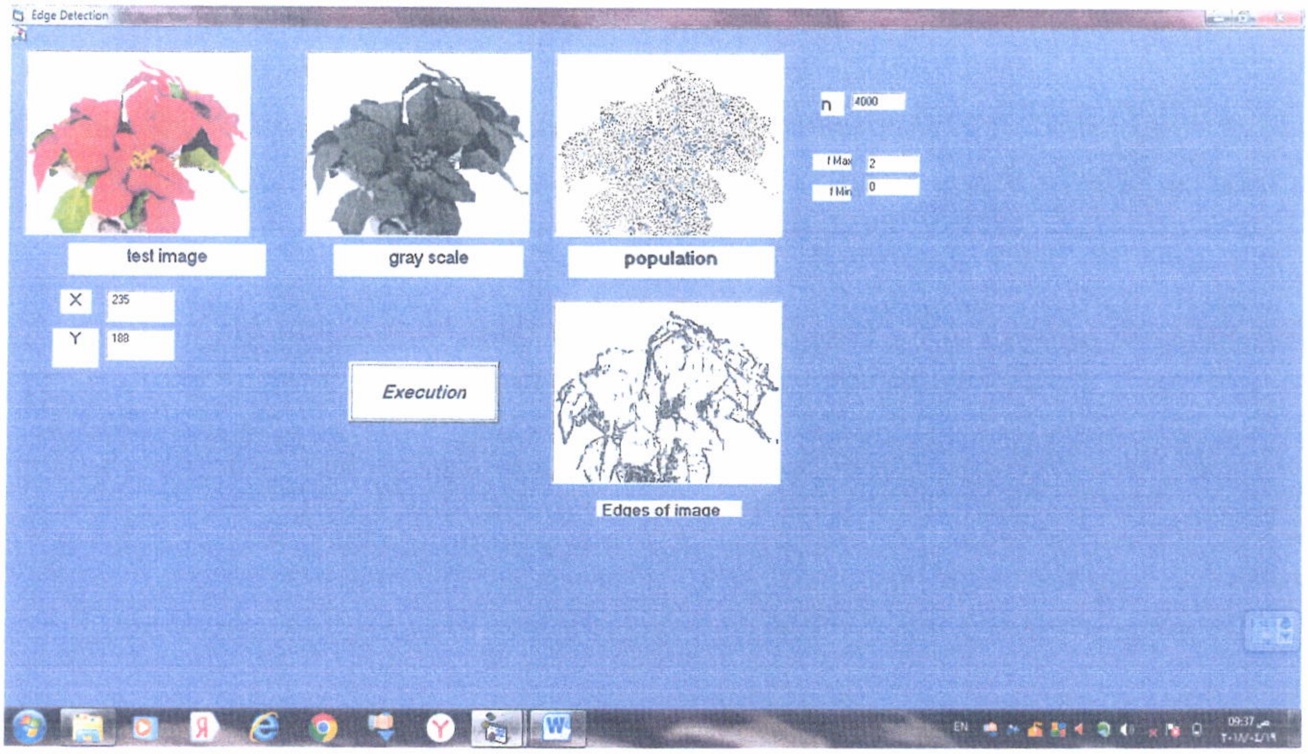
n 4000

I Max 2

I Min 0

Execution

Edges of image


Edge Detection
test image
gray scale
population
X
235
Y
188
Execution
Edges of image
n
4000
f Max
2
f Min
0