Namal Institute Mianwali
Computer Science Department
Artificial Intelligence
Fall 21

Document:	**Assignment/Project No.1**	**Date:**	**30-10-2021**
Prepared by:	**Jawad Ahmed (BSCS 2019-59)**	**Prepared for:**	**Dr. Junaid**
Document Details:	**This document contains my understanding and explanation of evolutionary algorithm in the light of 4 stages of the scientific process. It also contains the implementation of evolutionary algorithm in python.**		

Assignment:

Evolutionary Algorithm

Problem Statement:

Given a larger and a smaller template image, you have to write a complete Evolutionary Algorithm that takes the two images and solves the template matching problem, i.e. finds the coordinates or location of the smaller image within the larger image.

In simple words, I can say that one of the objectives of solving the above problem is that there are many linear search algorithms that may be efficient but these do not obey the principles of artificial intelligence. So in this problem of searching a template image in a group image we are actually practicing one of the concepts of artificial intelligence that is evolutionary algorithms.

Another objective is that we should follow the scientific process (observation, hypothesis, experimenting, analysis, modifications and results) to solve the problems keeping in mind the artificial intelligence concept i.e evolutionary algorithm.

Supervisors:

- **Dr. Junaid**
    - junaid.akhtar@namal.edu.pk

1 Abstract / Overview of Project:

1.1 Context:

- In the scope of this report, I will generally define the problem as such that I will find the best combination of element (template image) that maximizes some fitness function, and I will accept a final solution once I have either ran the algorithm for some maximum number of iterations or I have reached some fitness threshold. This scenario clearly uses the evolutionary algorithm which is based on **Natural Phenomena.**

1.2 Natural Phenomena:

- If we look around , evolution exists in almost everything. If we imagine the NAMAL 5000 years ago, there may exist an ocean because there are many fossil fuels found in the mountains of NAMAL which shows this area was covered with water many years ago but as time goes everything has evolved which brings variations.

- In the same way if we imagine the beginning of human life means Hazrat Adam (A.S) era. That time there were only two people which meant there was just one language to communicate. Presently, we all are their descendants. So in this way we should speak the same language spoken by Hazrat Adam (A.S) but there are opposite phenomena, I mean nowadays there are more than 6000 languages spoken around the world. The point is how do these languages come into existence? It was actually one language that was evolving time by time and brought variations in languages.

1.2.1 OBSERVATION:

- If we observe the language of two nearby districts, we notice there is minor variation in their languages. But when distance extends, if we take examples from two provinces like KPK and Punjab, there is a great variation in their languages like pashto and punjabi which are totally different. It is all because for the neighbouring areas there is small evolution in language which brings minor diversity but as the process continues for the large area distance people are unable to speak with each other because language is mostly evolved which is totally different from the first one.

- If we observe birds beak there is variation i.e in order to pick the insects from the island beak of finch (bird) is of different sizes. It is because different islands have different properties so therefore,

birds' beaks evolve in such a way that fit to that environment and can catch their prey easily for food (survival) as shown in Fig no.1. The evolution of the bird's beak to best suit(fitness) the environment is totally natural.

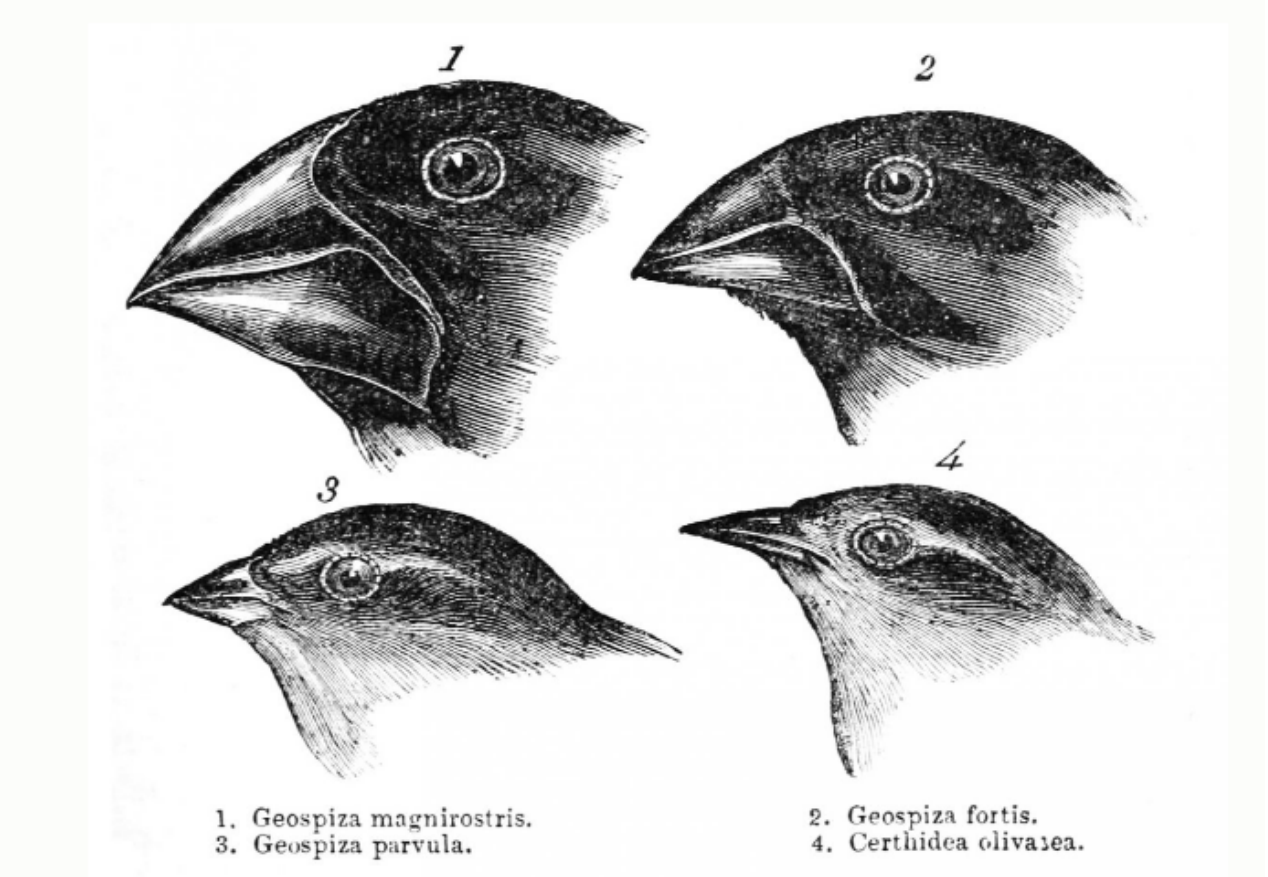


Fig no.1

- If we observe further around, there are many things which are evolved in order to the survival of the fittest.
- In the same way, if we deeply observe all the living organisms, these all follow the process for the survival of the fittest which was named evolutionary theory by the Darwin's.

1.3 Evolutionary Theory:

- Evolution means gradually developing.

- Charles Darwin's theory of evolution states that **evolution happens by natural selection [1].**
- Although it is true that we who believed in absolute power (God) existence, we mostly reject Darwin's Theory. But if we examine everything in depth, we will not consider Darwin's years of work as a waste of time. There might be some points which may not be acceptable from a religious point of view but there is a natural process which can not be denied like variation occurs due to crossover or mutation which leads to diversity.
- The theory of evolution demonstrates that all species in the universe exist due to alteration in their genes. The alteration in genes may vary according to situations because it depends upon natural factors like climate, temperature, areas etc. It was basically due to natural selection and their demand to live and survive in a better way. In order to come close to the fittest, small changes in phenomena need which can cause huge changes (evolution) but it may require a lot of time **as shown in fig 2 below.**

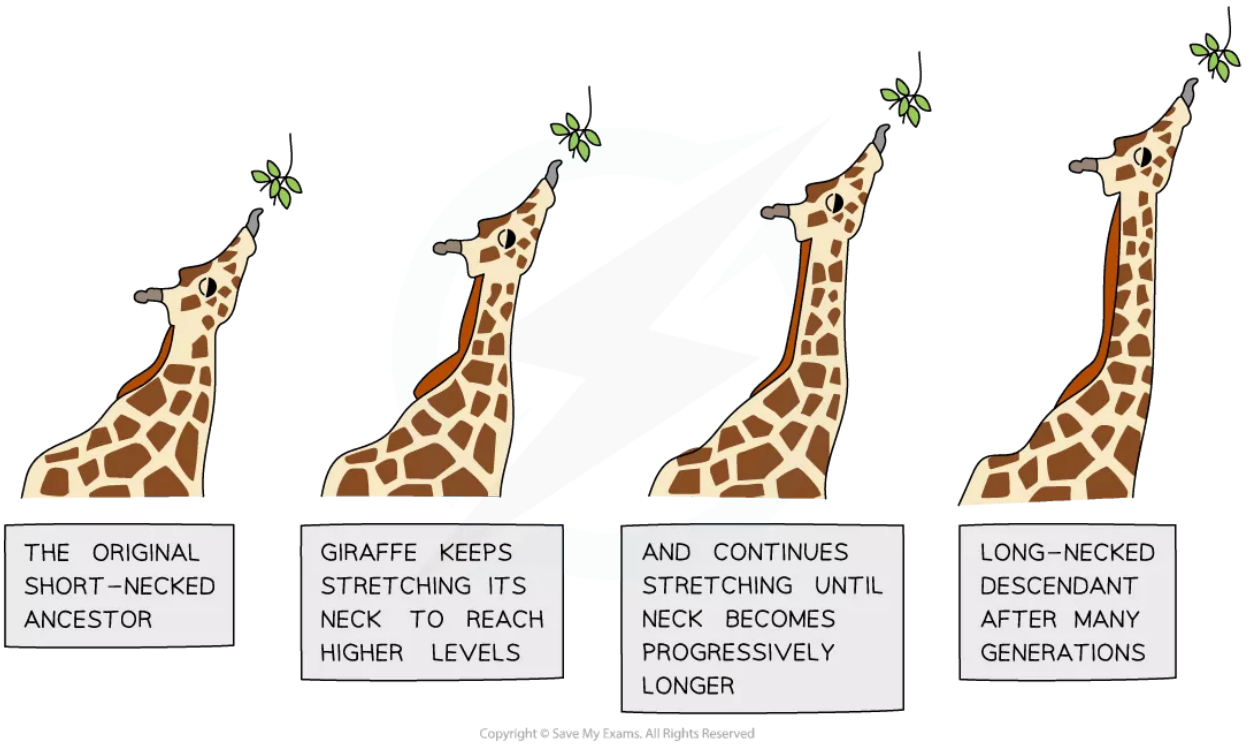


Fig no 2

- Nowadays, many problems are solved by this theory using computation which is named evolutionary computation. In order to do evolutionary computation many algorithms are used, one of them is evolutionary algorithm. In our context we use it for image recognition.

1.4 What is an evolutionary algorithm?

- Evolutionary Algorithm is inspired from Darwin's Theory Of Evolution.
- This algorithm is the mimicry of natural selection, where only the fittest individuals survive through the process of mutation, selection and crossover.
- This algorithm is a heuristic-based (use to make decisions) approach to solve problems.
- This algorithm is used to find, optimize and design for solving problems quickly.
- There is also randomized method use in this algorithm due to which it does not guarantee for the accuracy but it may be considered as approximately.

1.5 Characteristics of evolutionary algorithms :

- There are six main characteristics of this algorithm.
    - Representation
    - Selection
    - Recombination
    - Mutation
    - Fitness Function
    - Survival Decision

- **Representation:**
    - Its mean how to define an individual
    - It is the way to store optimization parameters.
    - It is determined according to the given problems because different problems have different requirements and parameters.
    - It has different types like binary representation, real valued representation etc.

- **Selection:**
    - It determines which individuals are chosen for the recombination and how many offspring each selected individual produces.
- **Recombination/Crossover:**
    - It determines how to combine the genes of selected parents.
    - Its type is determined according to the representation.
        - It may be bits of the genes
        - It may be Values of the genes
- **Mutation:**
    - It is used to change the gene/bit of every individual.
- **Fitness Function:**
    - It gives the intuition about how good the individual one is.
- **Survival Decision:**
    - It gives the idea of survival of the best individuals.

2 Implementation of Project:

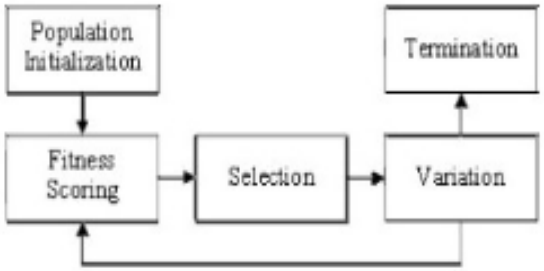
2.1 Use of Language:

- I use python language for the implementation of evolution algorithms to solve the given problem.
- I use some of the python libraries which provide ease in implementation like matplotlib, numpy, random.

2.2 Computational Model:

- In order to implement EA for the given problem I followed all the characteristics of EA as defined above. But I use each characteristic according to the requirements, explained below:
- First of all I follow the flow chart below given by the instructor.

Flowchart of Evolutionary Algorithms (EA)



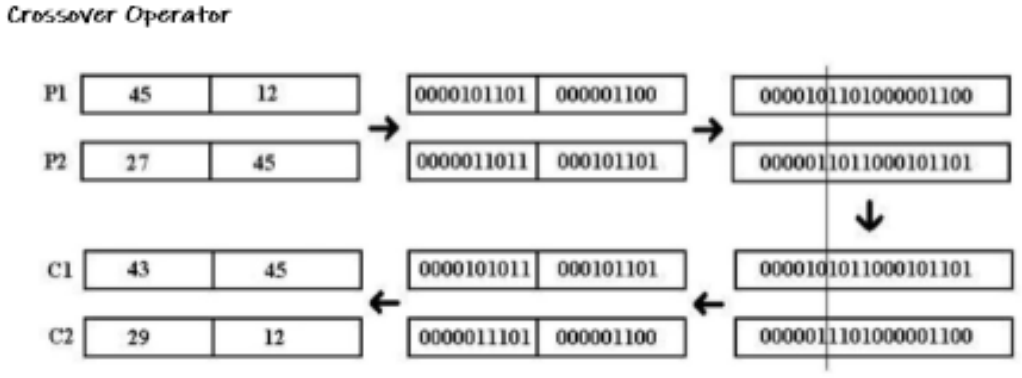
- **Initialization:**
  - I start with making random guesses, which means randomly generated points (x,y) for the possible solution and these guesses are called population. In this population I performed operations later on.
  - In order to generate a random population first I make a grid then I populate random points in the grid and later on use those points as shown below:
  -
```
def populInitialize(row, column, size):
    # Make Grid
    grid = np.zeros((row, column))
    population = []
    while len(population) < size:
        x, y = randint(0, len(grid) - 1), randint(0, len(grid[0]) - 1)
        population.append((x, y))
    return population
```

- **Fitness Score:**
  - In this function I pass an array of group image, template image and also random population generated in the initialization function. In this function I slice the group image upto template image dimension using the coordinates of the random population.
  - I use the following correlation formula in order to find the similarity between the template image and the randomly generated images in the group image.

```
# Finding correlation
for i in range(len(rndPopulation)):
    if len(rndImgs[i][0]) == len(targetImg[0]) and len(rndImgs[i]) == len(targe
        numerator = np.mean(
            (rndImgs[i] - rndImgs[i].mean()) * (targetImg - targetImg.mean()))
        denominator = rndImgs[i].std() * targetImg.std()
        if denominator == 0:
            corrVal.append(0)
        else:
            result = numerator / denominator
            corrVal.append(result)
    else:
        corrVal.append(-1)
corrVal = [round(num, 2) for num in corrVal]
return corrVal
```

- This function returns the similarity score mean correlation values.

- **Selection:**
  - In the selection function I passed a randomly generated population and also fitness score.
  - This function sorts the points according to the highest fitness score.
  - This function returns the ranked Population.
- **CrossOver:**
  - In this function I passed the ranked population.
  - First of all I convert the population into the binary because it is to do recombinations in binary.
  - Then I make the parents by the combination of two different tuples.
  - After that use the following logic given by the instructor to do crossover.



  - In terms of programming I use following method:

- ```
  # CrossOver Between X and Y Parents
  cross_point = random.randint(1, 5)

  for i in range(len(xyParents)):
      evol_XY_par1.append(
          [(xyParents[i][0][0:cross_point + 1] + xyParents[i][1][cross_point+1:22])])
      evol_XY_par2.append(
          [(xyParents[i][1][0:cross_point + 1] + xyParents[i][0][cross_point+1:22])])
  ```

    - After crossover I again convert the binaries into decimals and make tuples.
    - In the end this function returns an evolved population.

- **Mutation:**
    - In this function, I pass the evolved population.
    - I use the following statement given by the instructor in the mutation.

      Mutation

      Mutation is a unary operator that is applied to a single individual with a very small mutation probability. All it does is to flip a bit from 0 to 1, or 1 to 0, and hence produces a variation in the population. Together with crossover, it gives exploratory powers to an EA.
    - In the beginning I mutate each individual but later on in experiment I change the mutation occurrence.
    - As like in crossover I convert the decimal into binary, same here occurs but here I don't make any parents but directly mutate each bit of the individual one.
    - ```
      x = random.randint(0, 5)
      y = random.randint(0, 4)

      for i in range(len(xBin)):
          if xBin[i][x] == 1:
              xBin[i][x] = 0
          else:
              xBin[i][x] = 1

          if yBin[i][y] == 1:
              yBin[i][y] = 0
          else:
              yBin[i][y] = 1
      ```

- **Termination:**
    - Eventually, the algorithm must end. There are two cases where this usually occurs.
        - When the algorithm has reached some maximum runtime.
        - The algorithm has reached some threshold value we defined. For the given problem I used the termination state like if the threshold value is greater or equal to the given threshold value by the user then stop the program and give the result.
        - At this function the final solution is selected and returned.
- **Final Stage :**
    - In the final stage I also add some additional features like plotting graphs and also locate the template image on the group image.
    - After running the program, all the functions work accurately and it results in the form of plots where the rectangle shows the template image on the group image.
    - At this stage the implementation part is completed. Now it's time for the hypothesis and experimentation.

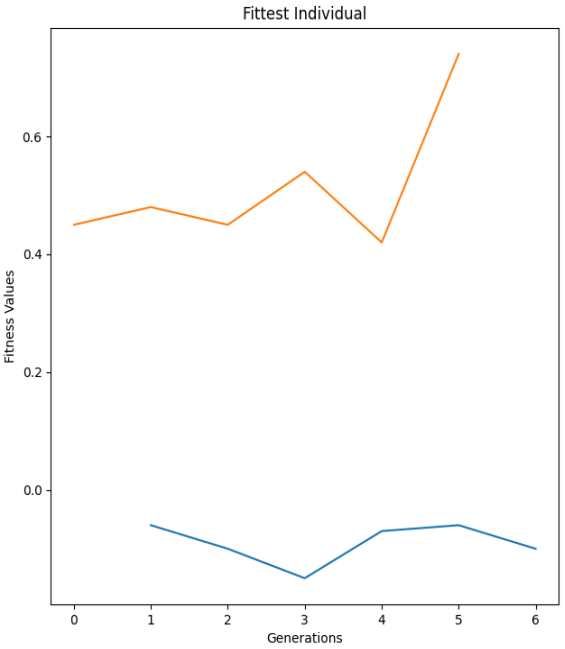
3 Experimentations:

3.1 Hypothesis No.1:

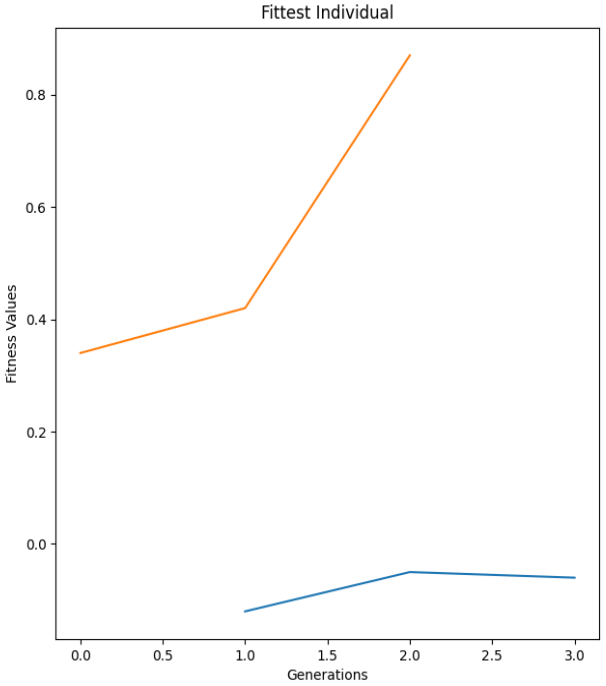
- There is a high chance of finding a threshold value for a given image, with the increase of generations. Means by changing the generation size but not changing the threshold value and population size what will be the result?
- The following experiment will test this hypothesis.

3.1.1 Experiment No.1:

- **1st Attempt:**
    - I keep the population size 100
    - I keep the Generation upto 500
    - I keep the threshold value 0.6
    - **Results**:
        - Following are the results on the basis of above data.

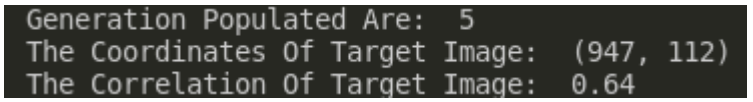


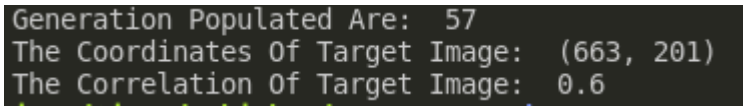

```
Generation Populated Are:  35
The Coordinates Of Target Image:  (341, 102)
The Correlation Of Target Image:  0.7
```

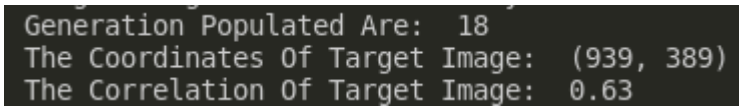



```
Target Image Found Successfully
Generation Populated Are:  2
The Coordinates Of Target Image:  (638, 106)
The Correlation Of Target Image:  0.87
```

- **Analysis**:
  - From the above results I analyze the following points.
    - Although I give generation value upto 500 but the termination stage is satisfied within 100 generations.
    - It gives quick results but our target is not achieved, in a very rare case the boothi is found.
    - After 20 or 30 or may be greater than these times running the program will give the accurate result i.e finding the boothi.

- Now in order to check that the above result is improving or not, I go for another attempt where I keep the generation upto 1000.
- **2st Attempt:**
  - I keep the population size and threshold value the same as the above attempt.
  - I keep the Generation upto 1000.
  - **Results**:
    - Following are the results on the basis of above data.
    - 

    - 

    - 

  - **Analysis**:
    - From the above results I analyze the following points.
      - Now if we compare this attempt with the previous attempt, one thing is guaranteed that if the threshold value is up to 0.6 there is a high chance that within 100 generations the termination stage will be met.

- It means if the generation size increases then the matching of threshold values also increases.

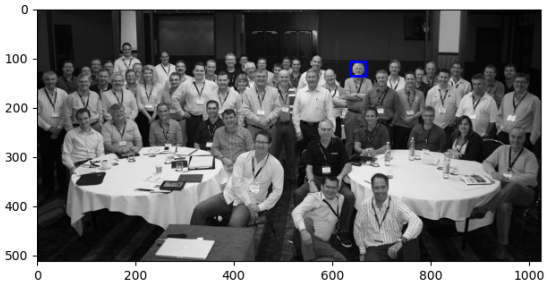
- **Conclusion:**
    - I tested hypothesis No.1 by experiment No.1. From this I came to know that chances of finding threshold value is increased with the increase in size of generation. It is because the algorithm is based on random generation. So randomly there will be a chance to find the result in the first 100 generations and there is a chance to find the result after 1000 generations.

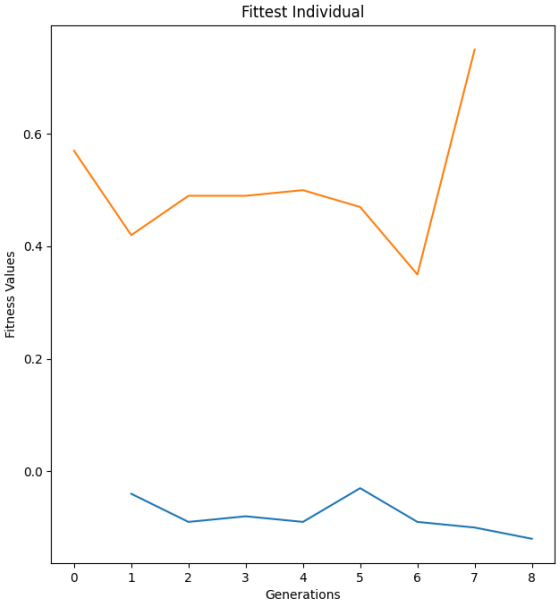
3.2 Hypothesis No.2:

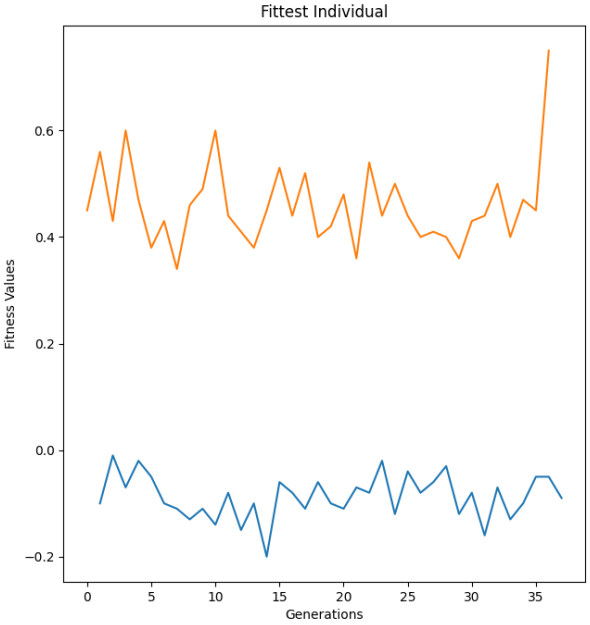
- By increasing the threshold value, there is a high chance of finding the accurate result and keeping the value upto 0.9 will give exact result means finding boothi in the group image but shall it be affected by increasing or decreasing generation size?
- The following experiment will test this hypothesis.

3.2.1 Experiment No.2:

- **1st Attempt:**
    - I keep the population size 100
    - I keep the Generation upto 500
    - I keep the threshold value 0.75
    - **Results**:
        - Following are the results on the basis of above data.





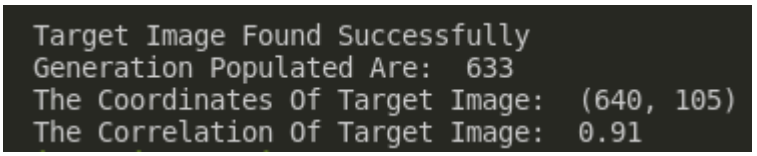


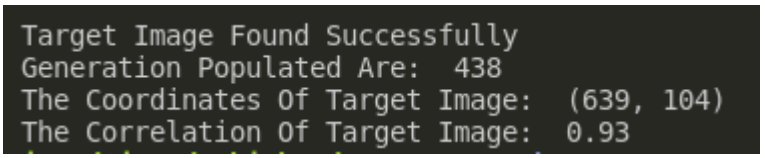
- **Analysis**:
  - From the above results I analyze the following points.
    - By keeping the threshold value 0.75 the termination stage is satisfied within almost 500 generations each time.
    - After 10 times running the program it gives me 4 times accurate result (find boothi) and 6 times not which means by keeping the threshold 0.75 it gives almost 40 to 45 percent accurate result.

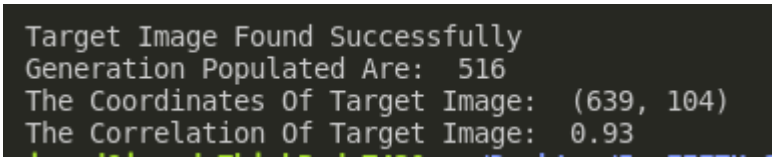
- In this case, as the threshold value is high so mostly boothi is found but not guaranteed each time in order to guarantee the accurate result I will keep threshold value 0.9 on next attempt.

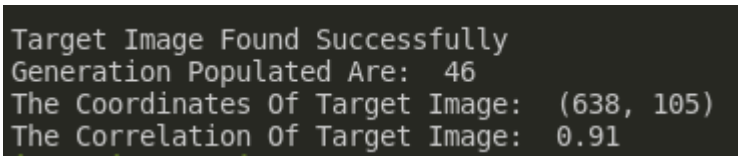
- Now in order to check that the above result is improving or not, I will go for another attempt where I keep the threshold value 0.9 .

- **2st Attempt:**
  - I keep the population size and Generation size the same as the above attempt.
  - This time, I keep the threshold value upto 0.9.
  - **Results**:
    - Following are the results on the basis of above data.









  - **Analysis**:
    - From the above results I analyze the following points.
      - On this attempt, one thing is guaranteed that if the threshold value is kept up to 0.9 then the boothi is must be found.

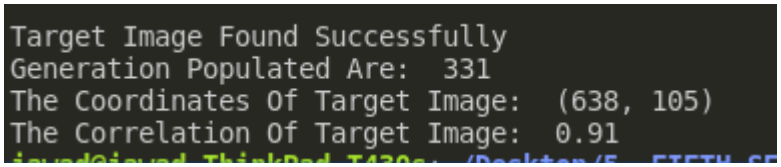
- After 10 times running the program, it gives accurate result upto 500 generation 6 times and 4 times gives result greater than 500 generations which means it gives 60 percent accurate result by keeping the generation 1000.
- On an average basis, almost upto 800 generations the termination criteria is met.

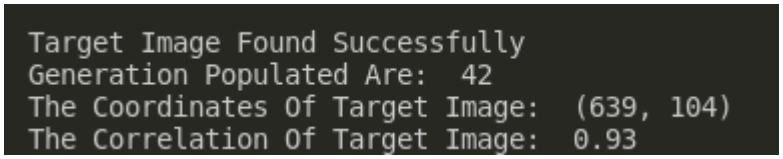
- **Conclusion:**
  - I tested hypothesis No.2 by experiment No.2 from this I came to know that chances of finding boothi is increased with the increase in threshold value.
  - Another result is that with the increase in generation size the probability of finding accurate results is also increased. Just like in the above attempt there is 60% percent chance of finding boothi for 1000 generations, so it will increase upto 97% if the generation size is kept upto 5000.

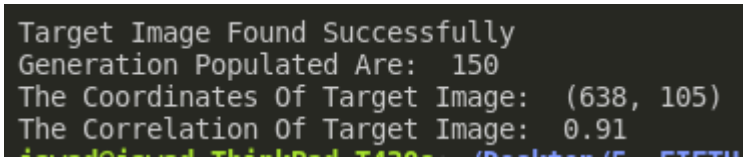
3.3 Hypothesis No.3:

- By increasing the population size the image recognition is also increased but it will take a lot of time because as the size of population is increased then the crossover and mutation time is also increased.
- Means what will be the results if the population size is changed but generations and threshold value is not changed?
- The following experiment will test this hypothesis.

3.3.1 Experiment No.3:

- **1st Attempt:**
  - I keep the population size 500
  - I keep the Generation upto 1000
  - I keep the threshold value 0.9
  - **Results**:
    - Following are the results on the basis of above data.
    - 

    - 

- 

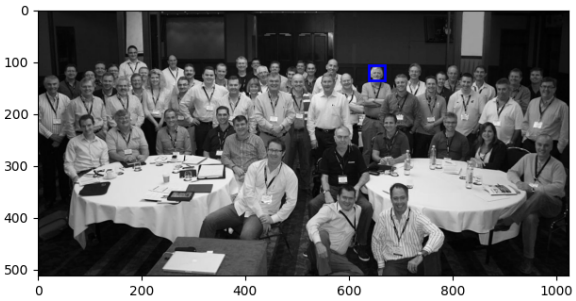
  - **Analysis**:
    - From the above results I analyze the following points.
      - By keeping the population size upto 500 then it will take a lot of time to find the accurate result.
      - As the above results shows that the termination state is met under 500 generations but I observed each time it took time by keeping population size 500.
      - In this case, after 10 times running the program, it gives accurate results upto 500 generations but taking time on each increasing generation means 150 generations will take less time as compared to 300 generations.
      - One thing is noted that to give an accurate result means finding boothi each time running the program.

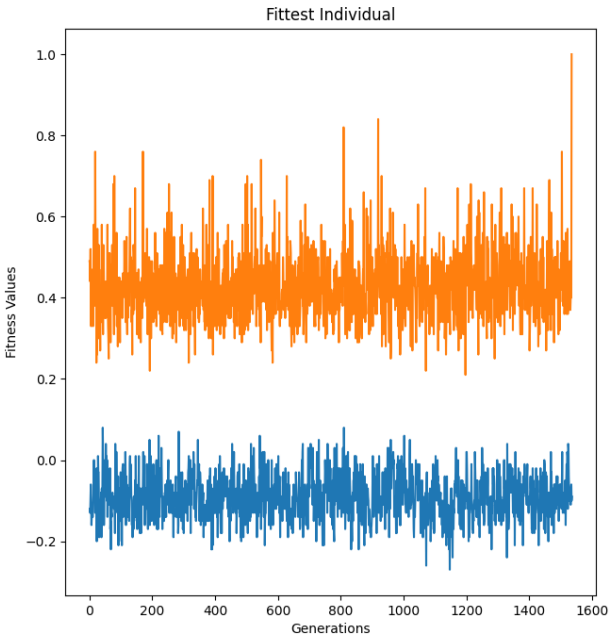
- Now in order to check that the above result is improving or not, I go for another attempt where I reduce the population size upto 50 .
- **2st Attempt:**
  - I keep the generation size and threshold value the same as the above attempt.
  - I keep the population size upto 50 in this attempt.
  - **Results**:
    - Following are the results on the basis of above data.
    - 

  - **Analysis**:
    - From the above results I analyze the following points.

- Now here is the very interesting analysis, which means by reducing the population size it will also reduce the chance of finding accurate results.
- In this case, after 10 times running the program, it is not giving me the accurate result.
- One other thing is noted that it goes to a terminating state fastly as compared to the previous one where I keep the population size upto 500.

- Here I go for another attempt that is by keeping the population size 50 but increasing the generation upto 5000 now let's test what happened.
- **3rd Attempt:**
  - I keep the population size and threshold value the same as the above attempt.
  - I change the generation size upto 5000 in this attempt.
  - **Results**:
    - Following are the results on the basis of above data.






```
Target Image Found Successfully
Generation Populated Are:  1535
The Coordinates Of Target Image:  (639, 105)
The Correlation Of Target Image:  1.0
```

  - **Analysis**:
    - From the above results I analyze the following points.
      - Now here I see that the boothi is found but it will take the generation upto 1500 which means if the

population size is reduced it will take to higher generation in order to find the accurate result.

- In this case, after 10 times running the program, it gives 60% accurate results having a very high value of generations.

- **Conclusion:**
  - I tested hypothesis No.3 by experiment No.3. From this I came to know that by reducing the population size the chance of finding the accurate result will also reduce because the population is concise due to which it is not evolved in such a way that it will go for an accurate result.

❖ **Close Mutation:**

- Now I close the mutation function and just try to cross-over. Now let's experiment and see results.

3.4 Hypothesis No.4:

- By closing the mutation, I think it will take a lot of time in finding the accurate results because just crossing over the fitness value will remain in the same range.
- The following experiment will test this hypothesis.

3.4.1 Experiment No.4:

- **1st Attempt:**
  - I keep the population size 100
  - I keep the Generation upto 1000
  - I keep the threshold value 0.84
  - **Results**:
    - Following are the results on the basis of above data.

  - **Analysis**:
    - From the above results I analyze the following points.
      - Here I noticed that by closing mutation and running a program by giving 1000 generations, it is not finding the result.

- Now in order to check that the above result is improving or not, I will go for another attempt where I increase the generation size.

- **2st Attempt:**

- I keep the population size and threshold value the same as the above attempt.
- This time, I increase the generations upto 5000.
- **Results**:
  - Following are the results on the basis of above data.

- **Analysis**:
  - From the above results I analyze the following points.
    - This time I noticed again that upto 5000 generations there is no result found.
    - But there may be an idle case where the accurate result may be found, it may be upto 10000 or greater.

- **Conclusion:**
  - I tested hypothesis No.4 by experiment No.4 from this I came to know that without mutation, crossover itself will not find the boothi in an optimal way.
  - So mutation is necessary for the optimal result.
  - Because without mutation the crossover is not evolving in such a way that leads to the accurate result i.e finding boothi in a group image. It is because after some generations the fitness values become constant due to which the boothi is not found.

❖ **Preserve The Fittest Values:**

- In order to improve the results from above experiments, I try to retain the fittest individual.

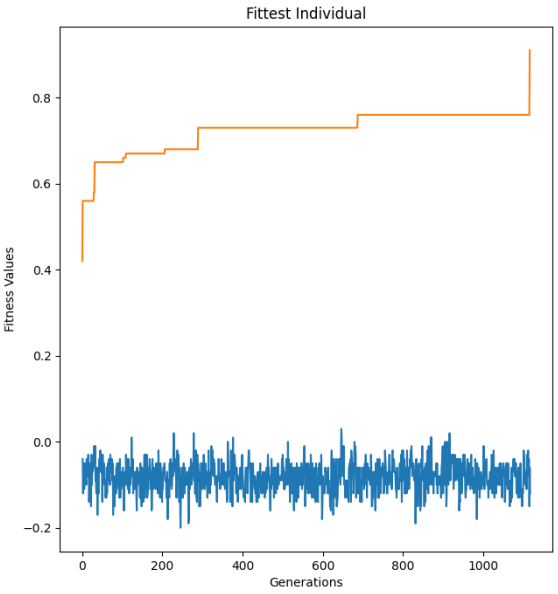
3.5 Hypothesis No.5:

- By keeping the fittest value preserved (not crossover and mutate) it may give good results.
- The following experiment will test this hypothesis.

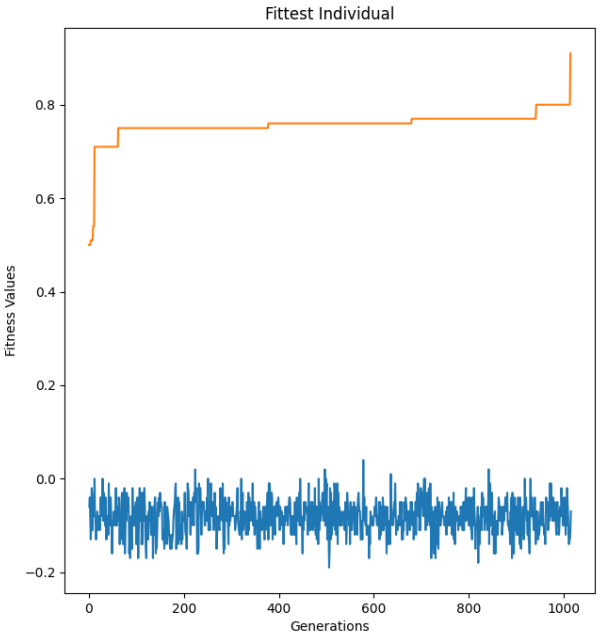
3.5.1 Experiment No.5:

- **1st Attempt:**
  - I keep the population size 100
  - I keep the Generation upto 5000

- I keep the threshold value 0.84
- **Results**:
  - Following are the results on the basis of above data.



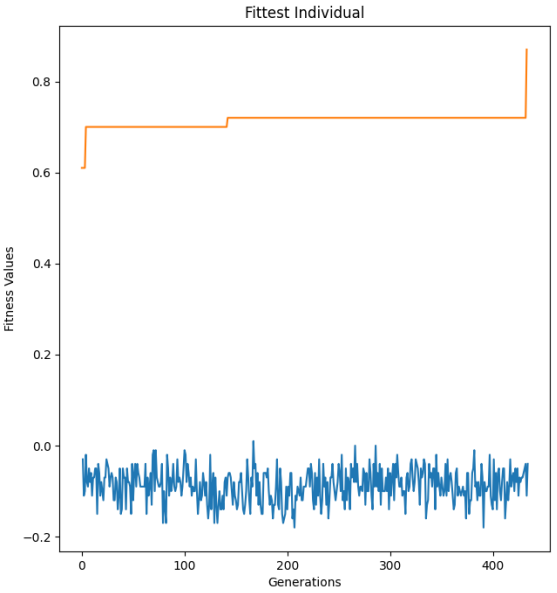

```
Target Image Found Successfully
Generation Populated Are:  1116
The Coordinates Of Target Image:  (640, 105)
The Correlation Of Target Image:  0.91
```



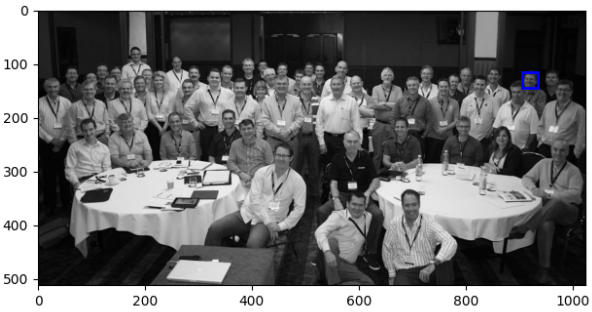
```
Target Image Found Successfully
Generation Populated Are:  1014
The Coordinates Of Target Image:  (638, 105)
The Correlation Of Target Image:  0.91
```

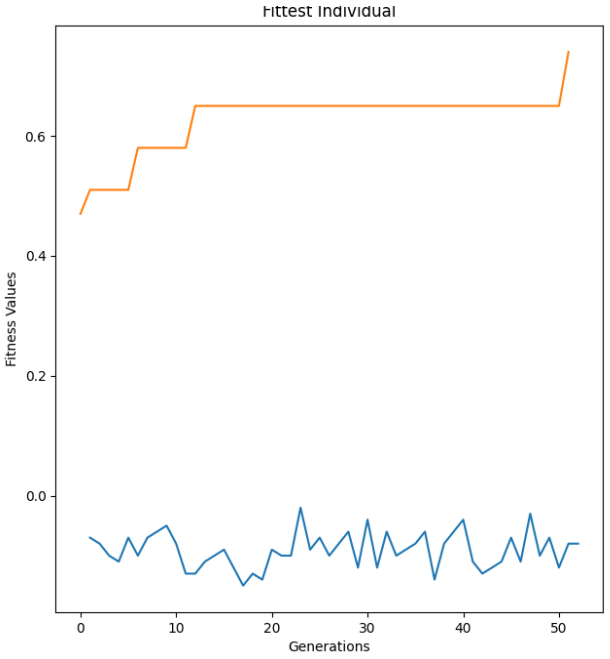
- **Analysis**:
  - From the above results I analyze the following points.
    - By keeping the threshold value 0.84 the termination stage is satisfied within almost 2000 generations each time, this time I also preserved the best fittest due to which in the above result I note that the fittest graph is increasing but not the mean graph.
    - After 10 times running the program it gives me accurate results but the generations exceed.
    - In this case, as the threshold value is high, boothi is found but not in the optimal way because it takes more generations to find the result.

- Now in order to check that the above result is improving or not, I will go for another attempt where I keep the threshold value 0.74 .
- **2st Attempt:**
  - I keep the population size and Generation size the same as the above attempt.
  - This time, I keep the threshold value upto 0.74.
  - **Results**:
    - Following are the results on the basis of above data.



```
Target Image Found Successfully
Generation Populated Are:  433
The Coordinates Of Target Image:  (638, 106)
The Correlation Of Target Image:  0.87
```






```
Generation Populated Are:  51
The Coordinates Of Target Image:  (905, 114)
The Correlation Of Target Image:  0.74
```

- **Analysis**:
  - From the above results I analyze the following points.
    - In this attempt, I observed that the terminating state is satisfied in a less generation but sometimes the correct boothi is not found because the threshold value is low.
    - After 10 times running the program, it gives accurate results within 500 generations which means in the previous attempt the generation exceeds the 2000 but the generation is controlled in this attempt although the probability of finding the boothi is also reduced.

- **Conclusion:**
  - I tested hypothesis No.5 by experiment No.5 from this I came to know that by preserving the best fittest value in the crossover will give little improvement then the previous experiments. Because the fittest individual is controlled in this experiment but as we see the mean of fittest is not controlled which means it needs a little more improvements.

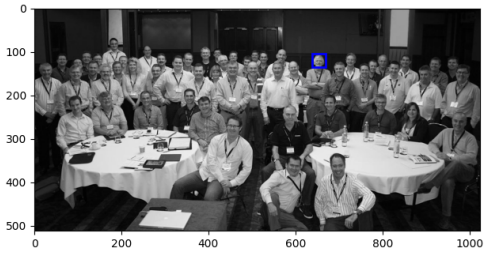
❖ **Now I do mutation on any random numbers:**

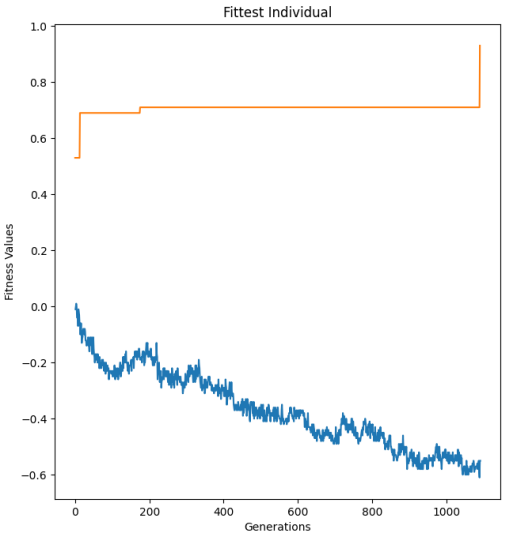
3.6 Hypothesis No.6:

- By keeping the fittest value preserved and doing mutation on any (0 to 5) randomly generated numbers will affect the mean graph of fittest or not. Let's test it.
- The following experiment will test this hypothesis.

3.6.1 Experiment No.6:

- I keep the population size 100
- I keep the Generation upto 5000
- I keep the threshold value 0.84
- **Results**:
  - Following are the results on the basis of above data.






```
ionaryAlgo.py"
Target Image Found Successfully
Generation Populated Are:  1090
The Coordinates Of Target Image:  (639, 104)
The Correlation Of Target Image:  0.93
```

- **Analysis**:
  - From the above results I analyze the following points.
    - Oh! This time the mean graph is down which means its randomness is controlled which was tested in previous experiment but it goes downward.
    - By keeping the threshold value 0.84 the image is found after many generations by mutating random numbers from 0 to 5.
    - After 10 times running the program it gives me accurate results but the generations exceed 1500 on average.
    - Again in this, as the threshold value is high, boothi is found but not in the optimal way because the mean graph is going downwards.

- **Conclusion:**
  - I tested hypothesis No.6 by experiment No.6. From this I came to know that by doing mutation on the basis of randomly generated numbers the mean graph is downwards.
  - The same result I found for mutating the least fittest individual which is not a natural process.

Problem not solved:

- I am facing a problem in setting up the mutation properly so that the mean of fitness value is control. But I will continue experiments until I reach the optimal solution of this problem.

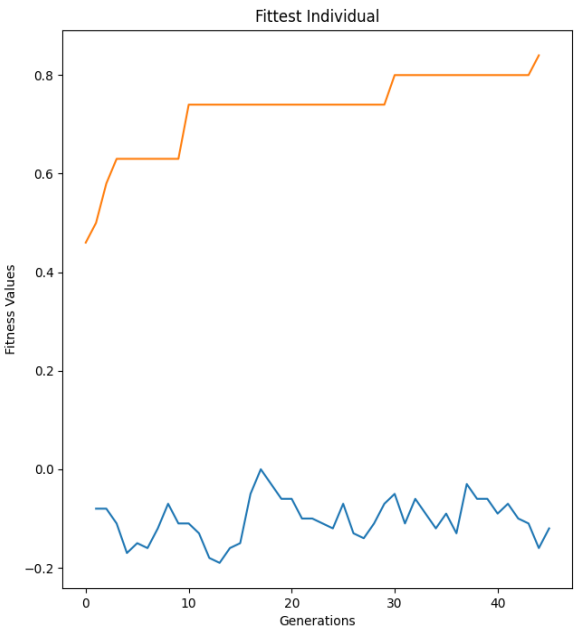
4 Limitations:

- There are limitations of the use of a genetic algorithm compared to alternative optimization algorithms:
  - This algorithm is based on randomness which is not guaranteed to give an accurate result in the same number of steps. On the first attempt it may find the result on 10 generations but next attempt it may find the result on 250 generations and so on.
  - There are a lot of computations occurring in this algorithm. Which means heavy computation machines may be required in order to solve complex problems with this algorithm.

- There is no assurance of convergence. There is a possibility that it gets stuck.

5 Conclusion:

- I have good experience by implementing one of the concepts of artificial intelligence because I learn a new concept that is how to solve problems with the artificial intelligence approach. One more thing, I feel that I am a scientist because I am observing, experenting and improving the results.
- While implementing an evolutionary algorithm, I faced a lot of difficulties in the beginning. But with the discussions and making logic for the specific tasks like how to initialize population, how to find fitness values etc it will lead me towards the final result.
- I try to improve the above experiment results as much as possible but I do not make the average fitness score graph in a linear way. I preserved the fittest individual and also tried to make the mutation less but unfortunately I did not get the mean graph straight, so there is need of improvement in this code.
- But mostly I got the accurate result (find boothi) upto 500 generations by keeping the threshold value 0.84 and population size 100.
- Here is one of the result:



Reference:

[1] https://www.yourgenome.org/facts/what-is-evolution