Evolutionary Algorithms

A project of Artificial Intelligence <3

By:
Ali Raza Khan
Saifullah Khan
Noman Shahzad
Sidra Ayesha

Acknowledgment

We are highly grateful to our respected teacher Dr. Junaid Akhtar, who provided us the opportunity to learn the scientific method and using the scientific method to solve computational problems. And we are on our way to becoming "scientists". :)

Problem statement

We have to find out a sample image from a big image, not by programming and engineering techniques but a scientific approach for solving computational problems. This technique is referred to as a different domain of algorithms called **Evolutionary** or **Genetic Algorithms.**

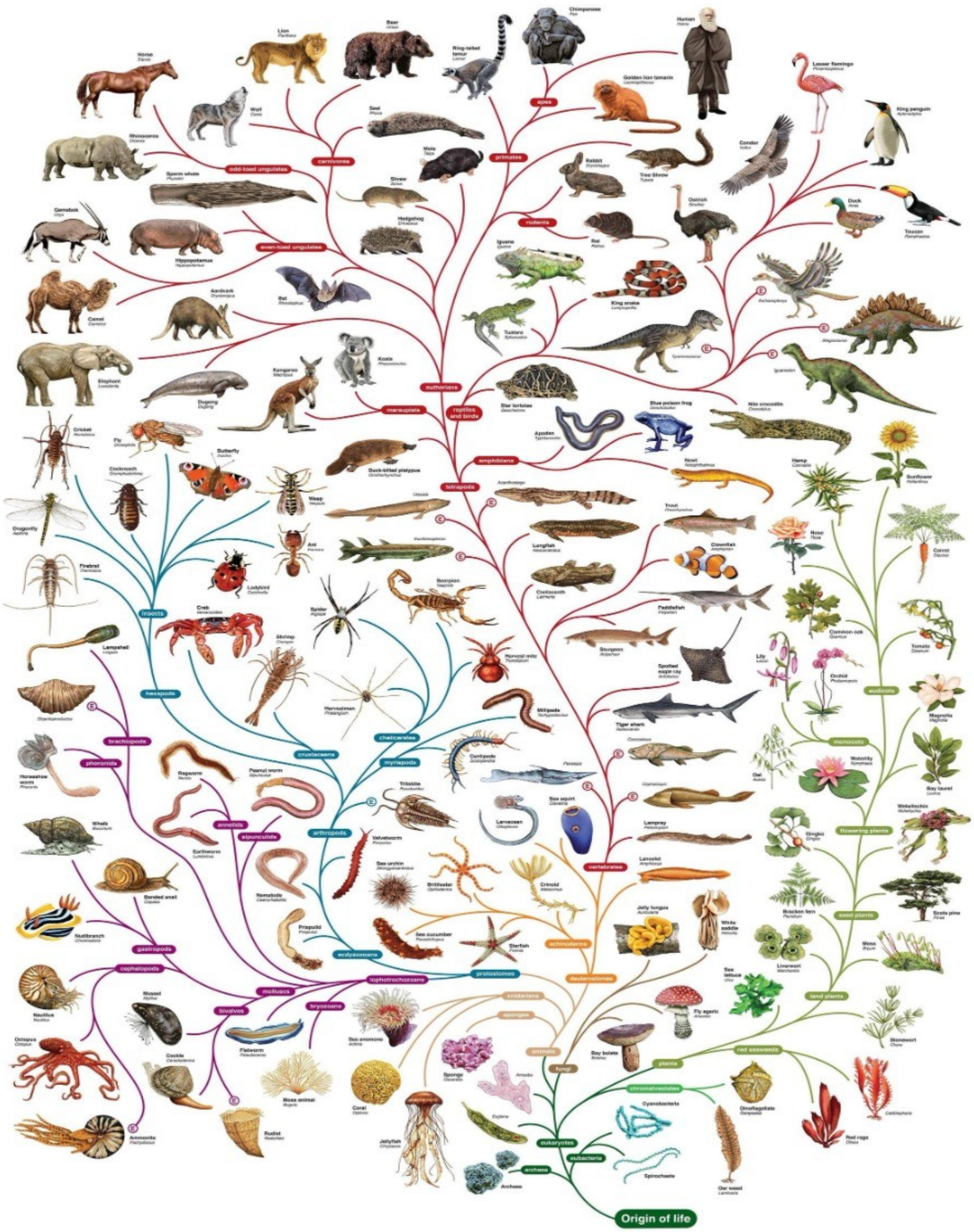
Our goal was just to find a specific car in traffic. Possible inputs are a picture of a group of many cars and a picture of the target car to be searched.

Natural Phenomenon+Theory:

The idea was inspired by a famous biological theory Theory of Evolution by Charles Darwin. He stated that "Every species is evolved from its ancestor species". He explained his theory by writing a complete book named "Origin of Species". The first thing which gained Darwin's interest was the beaks of Finches. He noticed that the same species, Finches, were having long beaks that were best suitable for eating grain, and in another region, Finches had short beaks, good for nut cracking. He studied it in his laboratory and came up with the idea that if grains are endangered then the finch with the long beaks would have low chances of surviving. But nature won't allow too quick change or evolution, but it would a species to whether to evolve or disappear slowly and gradually. Based on this he stated that specie is evolved from a single cell of origin, so are humans.

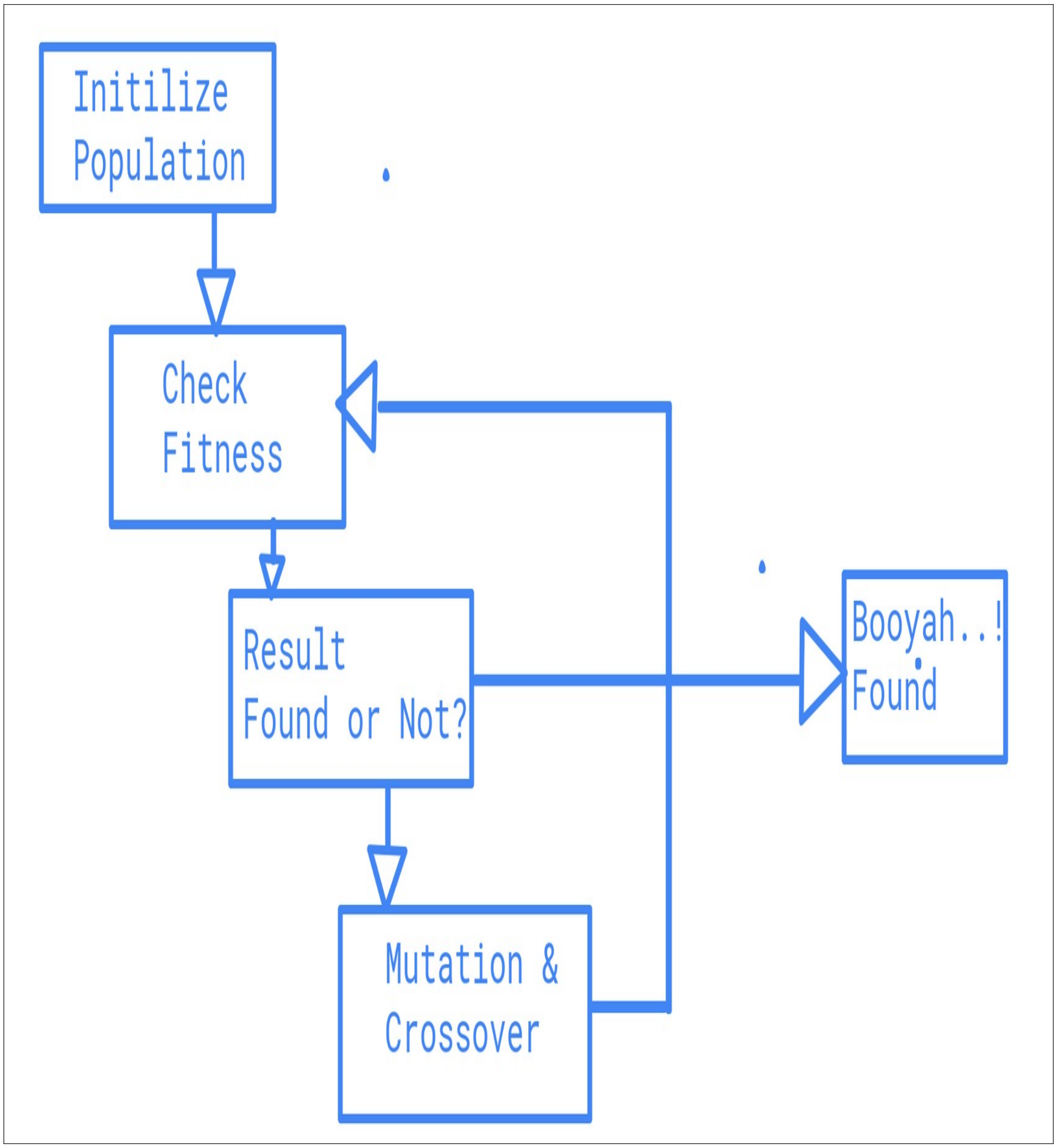
The key idea was that one species turn into another species with a very small change as time passes by, depending on the requirements of the environment. If the species can survive in that environment it is referred to as "Best Fit", also these two words are the base of his "Theory of Evolution".

Hence our algorithm uses the same idea of Darwin's theory of evolution, that our initial population can evolve into another with a very small change, but deriving from its parent. Hence we select a patch of the large image and compare it with a targeted search which will give us a constant matched value. And based on this value we add small changes to each member of the population. By juncture of two members having the highest matching values, we determine two new children which may take us closer to our desired output. Below is a picture of natural phenomena and the representation of what Darwin purposed in his books embedded in the document:



Computational Model:

Below is a graphical representation of the actual computational model we followed in solving the problem.



Application:

Following are the steps that we have used in our algorithm:
Initialize Population
Fitness

selection
crossover
mutation
stopping criteria

Initialize Population:

We initialized the population by generating some random points from the main large image. The maximum population was taken of count 40.

Fitness

In this step, we compared the small image with all points of the population. We found the correlation between small images & initial points. By this, we found that how much small image matches these points. And stored co-relation of all the points. Then we sorted the co-relation values.

Selection:

We selected values in pairs in accordance with co-relation values or matching percentages. For example, in a sorted array or list, we made pairs in descending order.

Check for Match:

Here we looked into our co-relation values and decided that whether we got the best match or not. The matching is done by a simple comparison with a value we call it threshold value.

Crossover:

In Crossover, we exchange some binary digits of sample1 with sample2 and the number of digits to be exchanged is defined by a random number. In this, we got two new children from two parents (sample1 and sampl2).
We are free to choose any of the crossover criteria. So, in our model, we first convert the X, Y points into binary numbers and then exchange some digits. After that, we reconvert it into decimal numbers as new X, Y points.

Mutation

In the mutation process, each sample undergoes a change of one binary bit, so to produce changes in the sample.

The Experiments...

Experiment-1. In our first experiment, we implement our co-relation function that takes two images of the same size as an input. The algorithm then compares these images pixel by pixel and count the number of pixels that matched with the target image. At the end of the iteration, we then calculate the percentage of matching (co-relation value).

This co-relation value is then stored against the X, Y coordinates of small scrapped images so that we can sort the population according to the matching values.

After sorting, we do a crossover. In the crossover, we first convert the X, Y coordinates into binary digits, concatenate them and then exchange some bits according to the value of the binary number. The exchange is done from the LSBs side.

When the crossover is done, the samples then go through the mutation process. A randomly chosen bit is flipped for these two samples. Mutation has many different criteria and we tried many of them.

We tried for 100% mutation.

Observation:

During our Exp-1, we didn't achieve the graph of rising co-relation value as the new generation is evolved.

Even we were not able to detect the exact or correct value of matching pixels.

Sometimes we get the image (target image) in less time while sometimes it takes too much time (like an exhaustive search).

Experiment-2. After Exp-1, we tried some new approaches in terms of crossover and mutations. We read articles and according to these articles the best rate of crossover is 0.95% and for mutation is 0.01-0.8% for a population of 40 (max range for best results and performance).

In the first Exp, we were trying 100% mutation and now we decrease it to 0.01% gradually and note the output in terms of co-relation value. We also change our mutation function. In Exp-2, we use the "match_tempelate" function of the Open Cv2 library.

But still the required graph was not achieved

Experiment- : In exp- , we implement some condition that is we will accept new offspring if its co-relation value is better than its parents if its value is less than the algorithm will loop until it finds better offspring. This will help us to improvise towards maximization.

One more interesting but annoying thing is that when we set the population range between - , we cannot find "baba ji ki bhoothi" even though the termination condition is reached. But when we increase the population up to or above we get the "bab ji ki bhooti" after some time (at max, we get "baba ji ki bhoothi" in generations starting from the initial population")

“We were not feeling good by putting a rectangle and to show our drawing skills, rather than the correct and obvious solution, so we did our best to implement different logic but none of them worked perfectly fine”

Here is link to our GitHub Repository in case if you are interested to have a look at all the methods we applied:

GitHub Link

OR

https://github.com/saifuullah/AI-FaceRecognation-project