Namal Institute Mianwali

Artificial Intelligence

Project # 01

Template Matching

Submitted To:

Dr. Junaid Akhtar

Submitted By:

Kiran Zafar

Acknowledgement:

I wish to express my sincere gratitude to **Dr. Junaid Akhtar** for solving computational problems using scientific methods. Furthermore, I sincerely thank their guidance and encouragement in carrying out this project work and helping me put the label of "Scientist" outside of my box.

1. ABSTRACT:

Template Matching is a high-level machine vision technique that identifies the parts on an image that match a predefined template. Template Matching techniques are flexible and relatively straightforward to use, making them one of the most popular methods of object localization. However, template Matching techniques are expected to address the following need: providing a reference image of an object (the template image) and an image to be inspected (the input image). We want to identify all input image locations at which the object from the template image is present. This document explains how the template matching problem is solved using a Genetic or Evolutionary Algorithm and describes the four stages of the scientific process, i.e., natural phenomenon, theory, computational model, application.

TABLE OF CONTENTS:

Contents

1. **ABSTRACT:** 3
2. **PROBLEM STATEMENT:** 5
3. **NATURAL PHENOMENON:** 5
4. **THEORY:** 6
5. **COMPUTATIONAL MODEL:** 6
6. **APPLICATION:** 7
   - (a) InitializePopulation(): 7
   - (b) EvaluteFitness(): 7
   - (c) Selection(): 7
   - (d) Crossover(): 7
   - (e) Mutation(): 8
7. **Experiments:** 8
   - 7.1 FIRST EXPERIMENT: 8

2. Problem statement:

The input of the program is a group photo and a single photo that is to be searched. First, we have to find an image from a group photo using Evolutionary or Genetic Algorithms, a scientific approach to solving computational problems.

3. Natural Phenomenon:

This idea is inspired by one of Charles Darwin's famous biological theories of evolution. He says nature decides which species or species of survival is prepared to survive in nature. "The generation of its ancestors produces every species," he explained. He explained this research theory in his book "Origin of Species."

The first thing to gain Darwin's interest was the beaks of Finches. Darwin used to be a sailor, so he noticed that the same species, Finches, had long beaks while Finches had short beaks in another region. Long beaks were best suitable for eating grain, and short breaks were good for nut cracking.

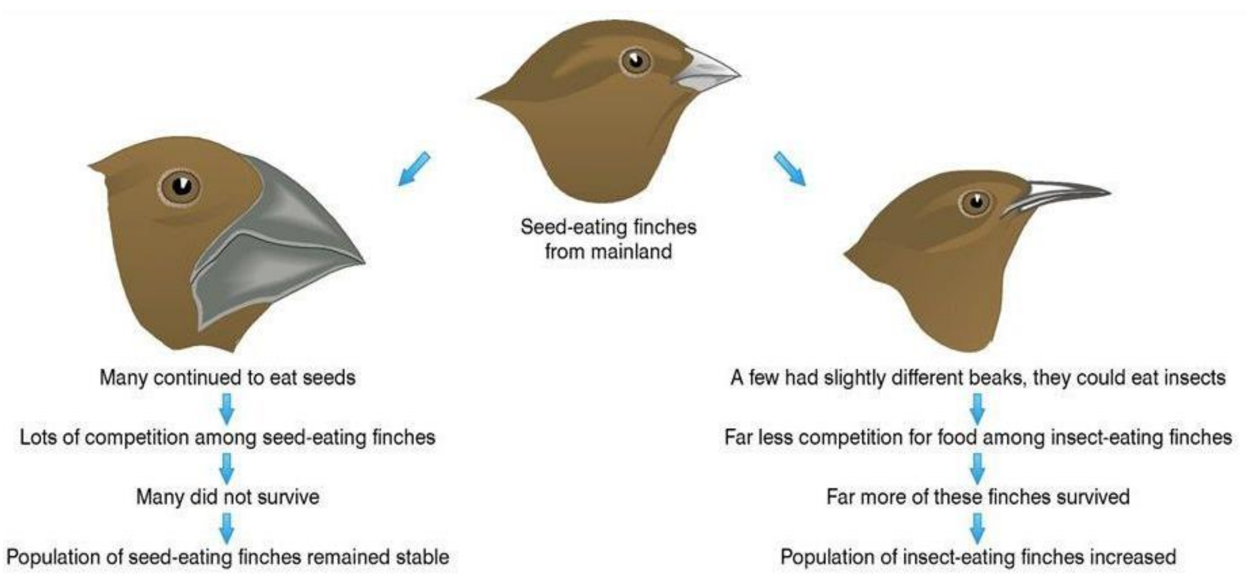


Figure 1: Natural Selection of Finches

So he studied it in his laboratory and came up with the idea that nature supports the survival of the fittest. If long-beaked Finches had to live in the short-beaked region, they would have died. At the same time, nature does not allow too quick change or evolution; it evolves or disappears slowly and gradually. Cutting a long story short, Darwin stated that specie is evolved from a single cell and evolved gradually influenced by nature to become human.

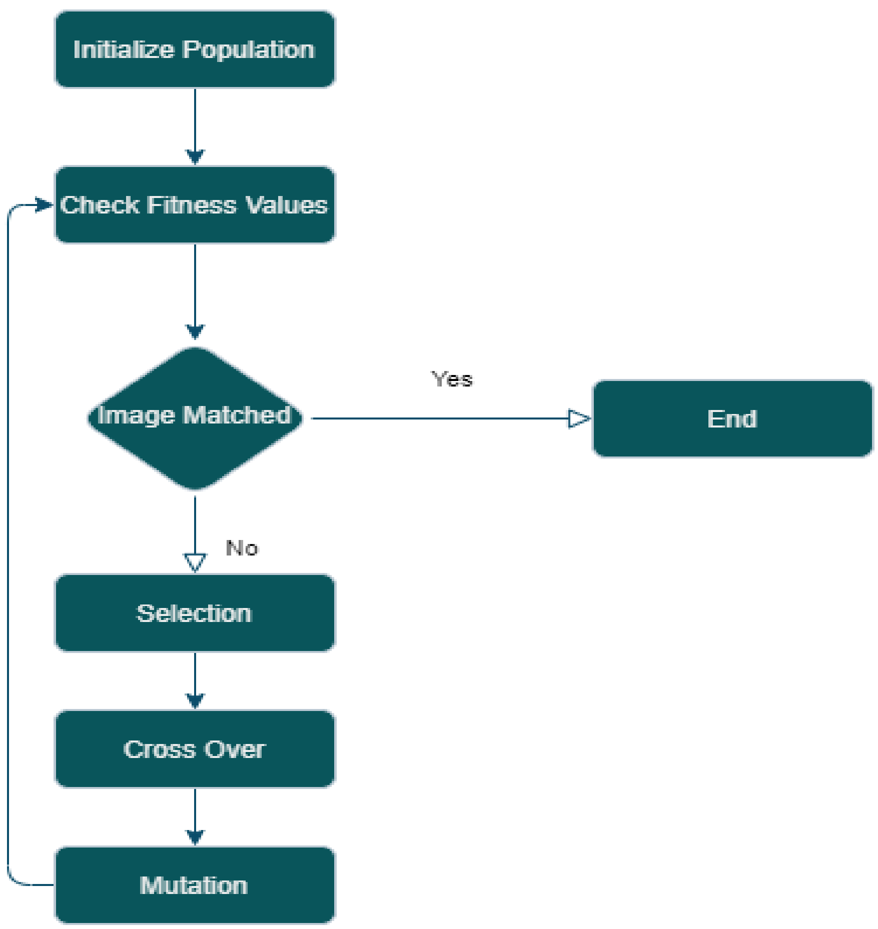
The main idea is that a species evolves itself to be the "Best Fit" in the particular environment to survive. These two words are the basis of his "Theory of Evolution."

4. THEORY:

Our algorithm uses the same idea as Darwin's theory of evolution. Our initial population can evolve into another with a minimal change, but deriving from its parent. Hence, we select a patch of the large image and compare it with a targeted search, giving us a constant matched value. Moreover, based on this value, we add small changes to each member of the population. Two members with the highest matching values determine two new children, which may take us closer to our desired output.

5. COMPUTATIONAL MODEL:

To solve the problem, following graphical representation of computational model is used:



6. APPLICATION:

Following functions/steps are used in implementing algorithm:

(a) InitializePopulation ():

From group/large image we generated some hundred random points to initialize population.

(b) EvaluteFitness ():

We compare the minor image points with all population points to find the correlation between the sample image and initial points. Finally, we come up with the solution of how much the sample image corresponds to these points. In this function, we also store all the correlation values.

(c) Selection ():

We sort co-relation values in descending order from which we get a ranked population.

(d) Crossover ():

For crossover, we can select any crossover criteria. One-point crossover is used; first, we select two parents and convert their x and y points to binary numbers. Then, we do slicing of [3:7] and exchange digits. After that, we reconvert these points into decimal numbers. By doing this, we get two new children (dec_c1x, dec_c1y) and (dec_c2x, dec_c2y). Following is the example of a one-point crossover.

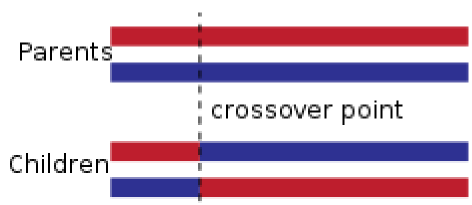


Figure 2: Crossover

Note: *At the 0th index, we have the fittest value, which can be used for the next generation. So, we do not use the first and last values (as we have a hundred values, and crossover can be done only on even pairs. We also have to discard the last value).*

(e) Mutation ():

In this case, we can also use any mutation criteria. So, in our application, the bit-flip mutation is used. In bit-flip mutation, we select one or more array indices and flip their values, i.e., we change 1s to 0s and vice versa, as shown below:

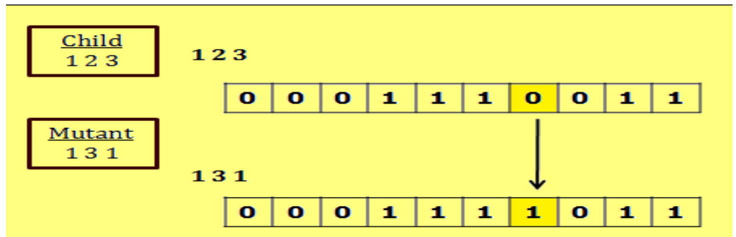


Figure 3: Flip bit Mutation

7. Experiments:

7.1 FIRST EXPERIMENT:

In our first experiment of a hundred percent mutation, we got our desired output four times out of ten times. It means sometimes we found "baba g ki boothi" or sometimes not. When we found "baba g ki boothi," output and graph of rising correlation values are given below:

Figure 4: Matched Image

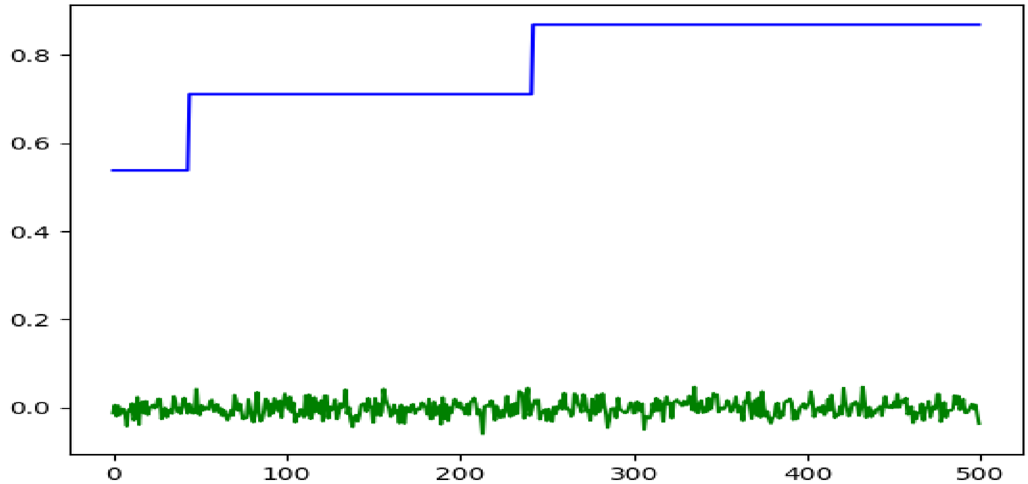


Figure 5: Matched Image Graph

However, when we did not find "baba g ki boothi," output and graph of rising correlation values are given below:

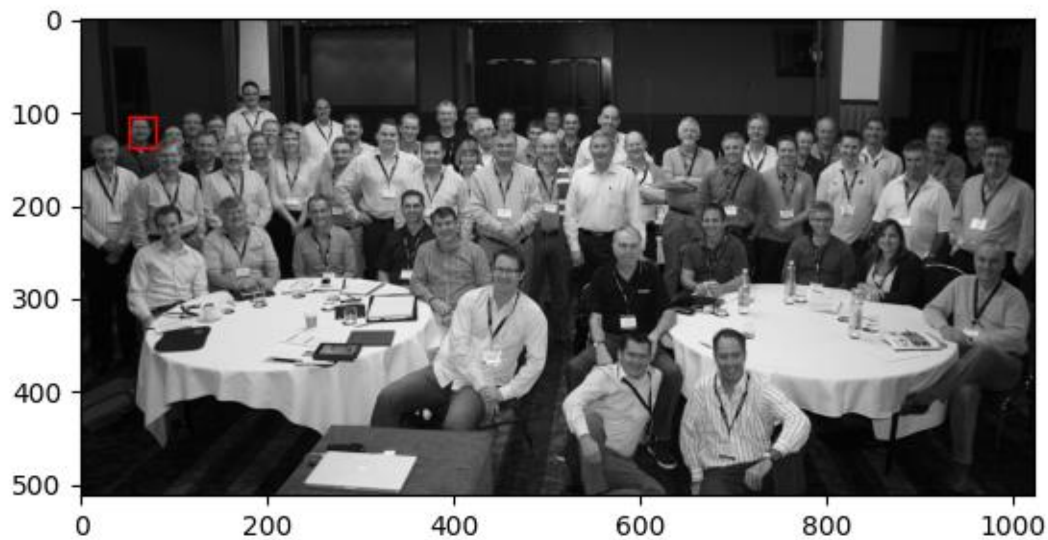
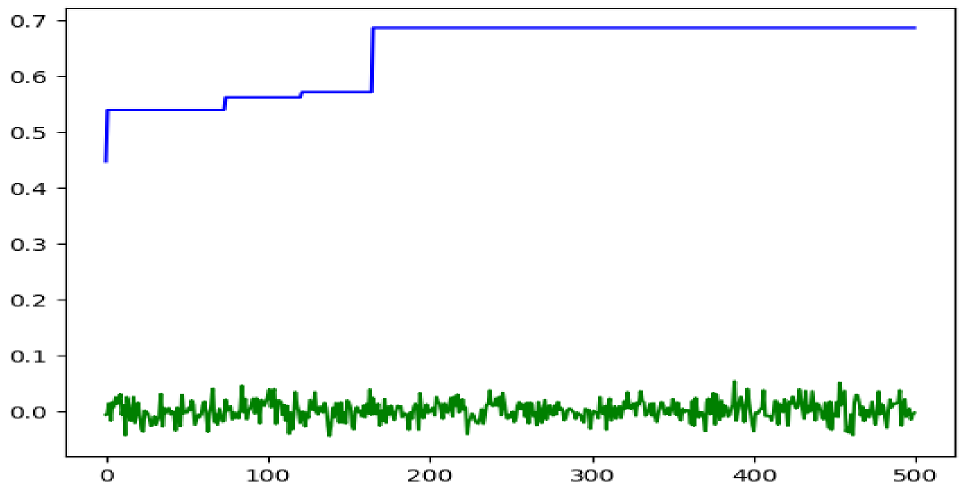


Figure 6: Non-Matched Image



Error:

Crossover rate generally should be high, about **80%-95%.** On the other side, mutation rate should be meager. The best rates reported are **0.5%-1%.** We were trying 100% mutation.