Visual Passwords using Genetic Algorithms.

Giri Gaurav Bhatnagar
Dept. of Information Science
PESIT
Bangalore, India
giri1gb@gmail.com

Kishan Kishore
Dept. of Information Science
PESIT
Bangalore, India
kishan2695@gmail.com

Aniruddha Hore
Dept. of Information Science
PESIT
Bangalore, India
being.anni@gmail.com

***Abstract*— A genetic algorithm is a search heuristic that mimics the process of natural selection [3]. This search heuristic can be used to generate images which are genetically similar to a target image. The genetic difference in the images gives us a way to use them as passwords. We start with a blank canvas and a target image and use a genetic algorithm to generate images which are genetically similar to the target image. Genetic similarity has been defined on the basis of a fitness function which compares two images pixel by pixel. These images of different genetic fitness values are then used to create a password mechanism to identify the target image where the image with highest genetic fitness corresponds to the correct password. The GA was also analyzed on the basis of parameters like time, fitness value etc. and relevant graphs were plotted.**

Keywords*— *Visual passwords,GA,Genetic algorithms,

I. Introduction

A genetic algorithm (GA) is a class of search algorithms which focuses on finding a good and robust solution which may or may not be the best. In this search it mimics the process of natural selection borrowing concepts like crossover, DNA, mutation etc. The algorithm starts by generating a random population. Next, the fitness of each member for this generation is calculated by applying a fitness function. A subset of this generation is selected and then mutated to form the next generation after which this process is repeated till the termination condition is reached.

The process can be used to generate images which are genetically similar to a given target image. The fitness function used here compares two images pixel by pixel. We start with a blank white image and try to generate images which are genetically fitter than the previous ones. The genetic difference in the generated images can be used as a property to distinguish them and use them as passwords.

II. Methodology

A. Selection of a suitable GA to apply on images

The first step in this project was to select a proper GA which could be applied (or modified) on image generation. [1] is an excellent python implementation of Evolisa [2] where a GA tries to generate images that resemble the famous painting "Mona Lisa" genetically. We re-implemented the GA making some minor changes in accordance to our requirements.

B. Generation of Visual Passwords

After the selection of the appropriate GA the next step was to generate images resembling a particular target using it. The algorithm iteratively generated images which evolved from a low fitness population to a high fit population. Images in every generation had different genetic fitness values as explained in the next section.

C. Analysis of the GA

In this step, the GA was analyzed on the basis of parameters like time taken to generate images and their corresponding fitness values. The comparison was done on different population sizes. Relevant graphs were plotted.

III. Algorithm and Implementation

The population size and the number of sides for a polygon are implementation specific. We chose the population size to be fifty and each polygon was either a triangle, a quadrilateral or a pentagon. Each polygon also had a RGBA [4] color value. The model considers an image (population) made up of polygons (individuals). We start with a image which is made of fifty randomly generated white polygons. This image serves as the initial population. The image which is to be used for generating password images is the target image. Fig 1 illustrates the GA used.

To calculate the genetic similarity of two populations we have a fitness function. Two images are said to be genetically similar if the fitness value is low, zero being the best possible value, which means that the two images are identical. There can be different methods to implement the fitness function. The most trivial being 'pixel by pixel' comparison. The fitness function used performs pixel by pixel comparison. d_{bi}, d_{gi}, d_{ri} are the differences in the blue, green and red attributes of a pixel $Pixel_i$, respectively, of the Parent and the Child image.

$$d_{bi}\,, d_{gi}\,, d_{ri} = ParentPixel_i - ChildPixel_i \tag{1}$$

(1) assigns the difference of the blue, green, red values of a pixel in a Parent image and a Child image.

$$fitness = \sum\sqrt{(\, d_{bi}^2 \; + d_{gi}^2 \; + d_{ri}^2)}\ . \tag{2}$$

(2) gives the fitness of an image.

```
parent = Generate a random White image, image  = Target Image
parentFitness = calculate the fitness of the parent image
while (current_iteration < MAX_ITERATION)
        child = mutate the parent image
        childFitness = calculate the fitness of the child image
        If childFitness < parentFitness
                parent = child
```

Fig. 1. The Genetic algorithm

The algorithm was implemented in Java and OpenCV [5] image library, was used for handling image processing related tasks. The program consisted of two classes – POLYGON and DNA – to represent the individuals and the population respectively. A POLYGON has a list of vertices and a RGBA color. A DNA represents a population and is made of fifty POLYGONs. When a DNA is mutated, a POLYGON of the DNA is randomly picked and either its color is changed or one of its vertex is changed as shown in Fig 2.

```
mutate(dna):
        polygon = pick a polygon randomly from the dna
        either changeVertex(polygon) or changeColor(polygon)
```

Fig. 2. Mutation in the GA

An initial DNA consists of 50 randomly generated white polygons. It is mutated and the fitness of the resulting 'child' DNA is calculated. This is then compared with the parent fitness. If we get a better child DNA, then it becomes the new parent DNA, otherwise the process is repeated. The flow chart for the algorithm has been depicted in Fig 3.

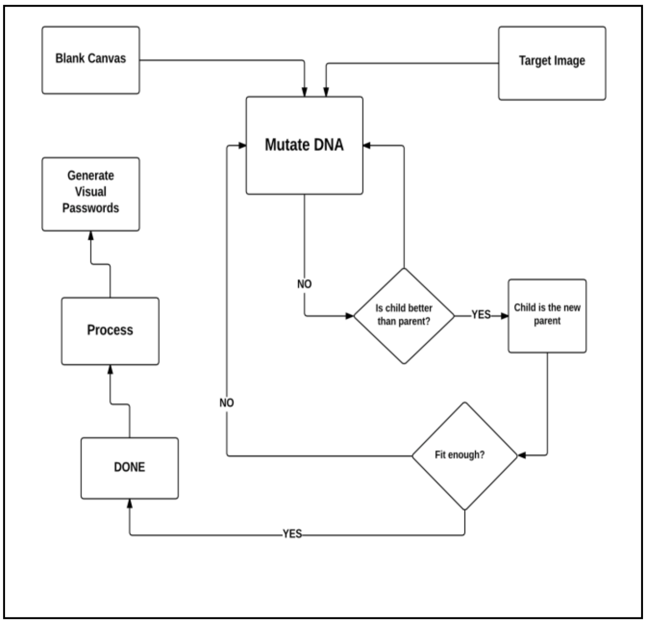

Fig 3. Flow chart

IV. RESULTS AND DISCUSSION

The result of our project consisted of several images arranged in increasing order of genetic similarity. This genetic similarity as explained in the previous section was based on a fitness function which compared the fitness of images to a target image. These images could easily be distinguished based on their genetic fitness as the ones with higher fitness had more resemblance to the target image. To use these images as passwords we used two different strategies. In both the cases, the target image was specified by the user.

In the first one, the user was shown 3 or more images having different genetic fitness corresponding to a single target image. The user had to identify the image with the highest genetic fitness to choose the correct password. By choosing images having little difference in their genetic fitness the difficulty of choosing the correct password could be increased.

The second approach was to present the user with 3 or more images (generated by our algorithm) but corresponding to 3 or more different target images. The user had to identify the image which corresponded to the correct target image. The difficulty in choosing the correct password could be increased by specifying very similar target images. The generated images would thus be very similar and could be identified only by the authentic user.

Fig 4 shows an image which was generated corresponding to the target image.

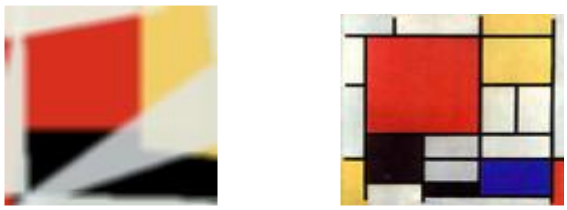

Fig 4. A generated image and its target image with a population size of 5 after 38600 iterations

The resemblance of an image that the algorithm (as specified in the previous section) and convergence towards the target image was dependent on various factors like the initial population size, the fitness function etc.

One major concern in this project was the time taken to generate these images which was quite large due to the inefficient fitness used in our algorithm which compared two images pixel by pixel resulting in inefficient comparison.

Fig 5 shows one of the many possible configurations of the images which could be shown to a user. The corresponding target image is the same as above.

Fig.5. A possible configuration of 3 images having different genetic similarity

When compared with the target image the third image seems most similar to it and thus is the correct password. The three images can be easily distinguished on the basis of their genetic similarity and act as an appropriate password mechanism.

During the analysis of the GA, its performance was observed starting with different population sizes. For a given target image, fitness values of the images generated by the GA, the corresponding time taken and the generation number were recorded. Graphs were plotted using this data which revealed some interesting conclusions. Fig. 6 shows a graph in which fitness value has been plotted against time for different population sizes for the same target image in Fig. 4.

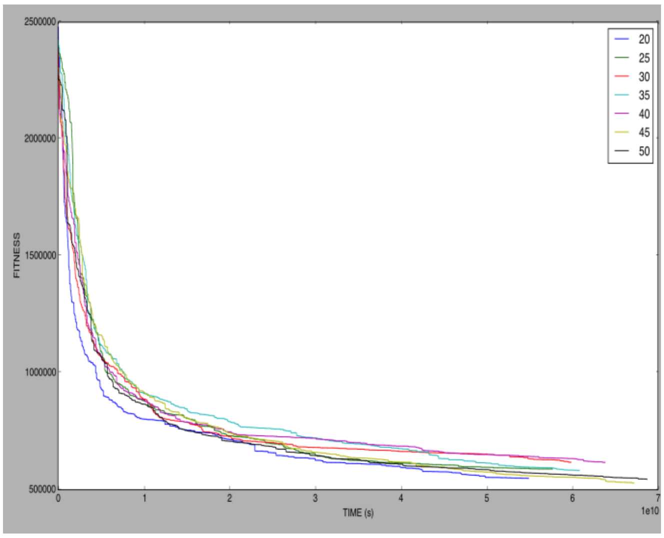

Fig. 5. Fitness value versus time

This figure shows us that for a population size of twenty the GA takes comparatively less amount of time to achieve a better fitness level. Fig. 6 shows a graph in which the time taken to generate the image has been plotted against the generation number. Fig. 7 shows a zoomed view of this image where the difference in time taken to generate consecutive images has been plotted against the generation number. It can be inferred from Fig. 6 and Fig. 7 that the fitness function used by the GA compares any two images in constant amount of time given they have the same population size.

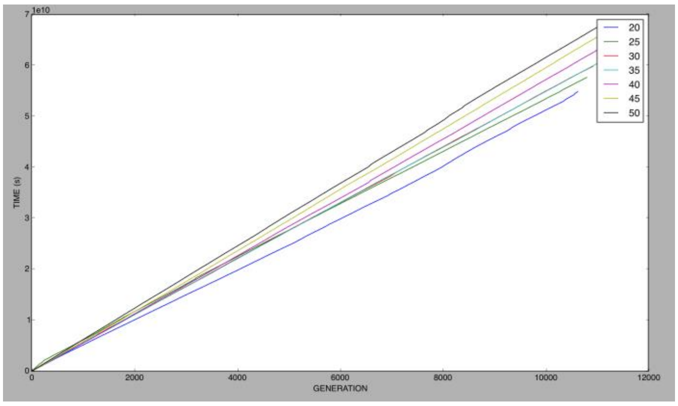

Fig. 6. Time versus Generation number

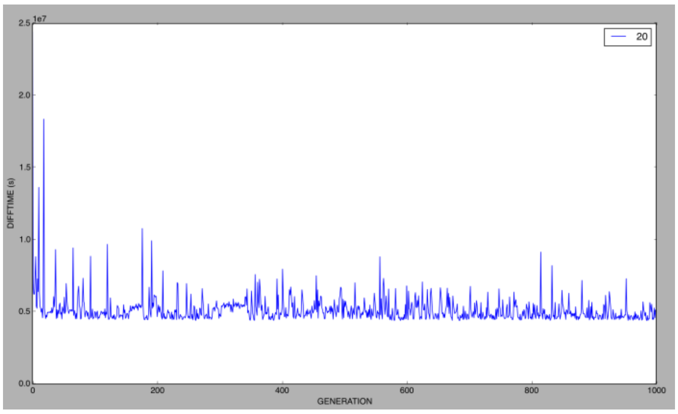

Fig. 7. Time difference versus Generation number

V. Conclusion

Genetic algorithms are a class of search algorithms which provide optimal solutions while mimicking the process of natural selection. They have a wide range of applications [6]. The application of genetic algorithms on images was studied in this paper. A target image was specified and then the GA was used to generate images which are genetically similar value to it. A fitness function which compared two images pixel by pixel was used to determine the genetic fitness value. The pixel by pixel comparison was a slow process slowing the down the overall process of image generation. The difference in the genetic fitness values of the images was served a property to distinguish between them and use them to create a password mechanism.

VI. References

[1] https://github.com/miezekatze/evolisa/blob/master/README.me

[2] http://rogeralsing.com/2008/12/07/genetic-programming-evolution-of-mona-lisa/

[3] https://en.wikipedia.org/wiki/Genetic_algorithm

[4] https://en.wikipedia.org/wiki/RGBA_color_space

[5] http://docs.opencv.org/2.4/doc

[6] https://en.wikipedia.org/wiki/List_of_genetic_algorithm_applications