Introduction to Artificial Intelligence

Assignment 2 | Selina Varouqa | BS18-02 | Spring 2020

Algorithm Representation

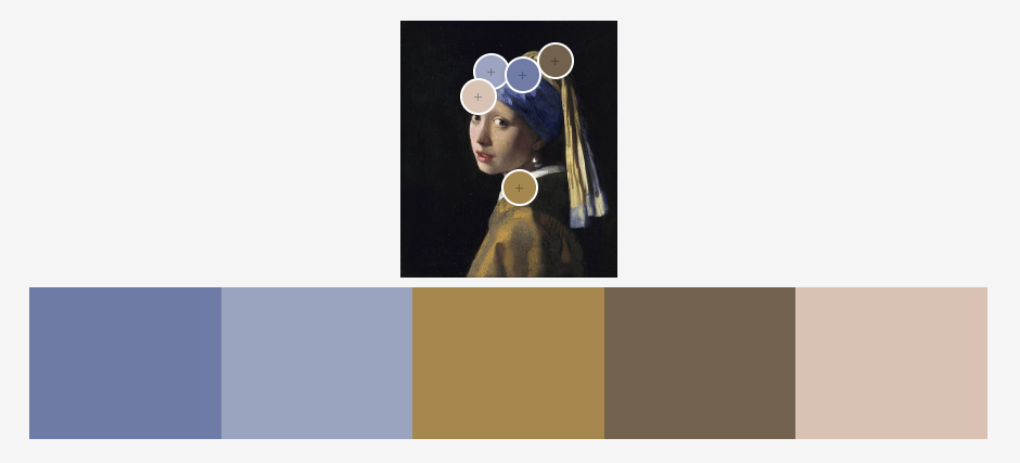
Genetic algorithms use linear binary representations. In my genetic algorithm I have used mainly arrays and data structures to represent the genes that I wanted which are equal sized squares.

my squares have the following attributes:

- they are located randomly on the canvas
- I specified the height and weight taken as parameters (same for the squares)
- all points are mathematically calculated based on the random start point which is the upper left point of the square by adding and subtracting width and height as the following:
    - upper left (x0,y0): completely random points which specify the location
    - upper right(x1,y1): we have to add width to x0 to get x1, and the y1 is same as y0
    - lower left(x2,y2): x2 is same as x0, but we subtract height from y0
    - lower right (x3,y3): x3 is as same as x1, and we subtract height from y1
    - **Note:** if I decided that I need a rectangle then I need the above four tuples but if I decided I need a square then I need only (x0,y0) and (x3,y3)
- since I am using RGB (Red, Green, Blue) type, then I have specified the colors to be random by randomly choosing the numbers between 0 and 255.

  then I decided to have randomly generated colors by 50% and the other 50% chance is from the set of the most commonly used colors extracted from the image. The goal of it was to make less color changes and speed up the process even if a bit to approach the final image. The set was generated by the following steps:

    - I inserted the target image in https://color.adobe.com/create/image and extracted the main colors (and some variations of them):



    - I extracted the HEX code, and then converted it to RGB, then created a list that contains the three RGB values for each color.
- the position of the square is set randomly set but with the bounds of the

I drew the polygon (the rectangle in my case) using `draw.polygon` and specifying the points of the corners and the fill colors as specified above.

I did not only use my rectangles, but down later it will appear that I have used image pixels manipulation as well. (to be explained)

Selection Mechanism

After calculating scores that we get from the fitness function, the selection goes by selecting top n scores from the scores that I have, so it is sorting and slicing the top n (specified by me).

Image Manipulation Techniques

1. **Opening the target image**: I open the final image with the `plt.imread()` function
2. **Converting the target image to RGB**: I check if I have in image in greyscale(2 channels) then I convert it to RGB by adding one channel and reidentifying channels in greyscale
3. **Drawing squares**: as explained in the previous section

Fitness Function (SAD)

my fitness function is the sum of absolute difference between the two images. It measures the similarity between image blocks, we can see it by the following formula:

$$\sum_i^p \sum_j^q |Pi - Qj|$$

Crossover Function

I have used Uniform Crossover Function as a start, where each gene is selected from one of the corresponding genes of parent chromosomes.

On the other side I also have used a a crossover function where I implement Random-Point_Crossover and this is like n-points crossover except that n is random and I switch a horizonal line instead of a uniform region.

Mutation Criteria

Mutation for me was changing the size of the squares or rectangles corresponding to the number of generation I have.

for example

`mutated_one = mutate(child, rectangles=4, width=50//(i//20+1), height=50//(i//20+1))` where `i` is my number of generation.

Perception of Art

> ***"Art should comfort the disturbed and disturb the comfortable."***
>
> *— Cesar A. Cruz*

I have always agreed with this saying. For me, art should make us feel something! Either good or bad, intense or shallow. The second something visual that is created by another person can affect your emotions, it is considered art from my own perspective. That applies to all forms of art such as music and theater. If we talk especially about visuals, I think the combination of those colors, lines, and so on (or even the lack of them) can make you feel something, or remind you of something, or provoke your thoughts, then that's the art of it! That is why throughout history, art has been subjective.
Speaking of the feelings, I believe that there is something called visual satisfaction, it might differ from one person to another, but at the end we will have common grounds as human beings such as seeing something familiar, or the Golden Ratio, or comforting colors of the nature, different combinations of them that can create that comfort to our eyes.
Combining that comfort and the feeling of something, and if some work can make us feel both, the I think it is an art.

Artistic Aspect

The target image that I wanted to recreate is **girl with a pearl earring** for the famous artist **Vincent Van Gogh** , I consider this painting pure art because what I have described in the previous section.

However, the art that I aim to with my genetic algorithm, is more of an art that creates a sense of familiarity, or resemblance to the image, even if it is just the colors or the shapes, since this is a combined effort between a human and an AI.

The outcomes of my algorithm start more as an abstract art that do not resemble to my target image but still considered as art, and other ones in more high numbered generations that you can see some corners of colors that correspond to the target image.

What can be improved to reach an accurate result?

1. fitness function: I think there needed a better fitness function, but it would be highly computational
2. the values of parameters, since our values of parameters should depend on our target image, more trial and error.
3. definitely much more generations, the most generation number I have reached is 2000. and I think it should be much more