2024

Applied AI Image Reconstruction Using a Genetic Algorithm

COMP 3710 - 02

LAURA LEFEBVRE AND INÊS MARIA

Table of Contents

Problem Description

Our project aims to create an artificial intelligence that will train to take and recreate an image, starting with simple two-colored images and possibly into grayscale images. We will do this with a genetic algorithm based on Charles Darwin's theory of natural evolution, where the AI will learn from mistakes and become stronger by creating the image faster and faster as it learns.

We chose image generation because it is so widely used by many individuals for many different reasons, whether for good reasons, like creating posters for events, or some wrong reasons, like creating images of other people. Image generation is also exciting because of how much training goes into making image generators more and more accurate in such a short time. It is a big topic with several different approaches to creating an algorithm to create an image of your choosing, with little to no effort from the user.

Proposed Method

The project aims to develop a Python program capable of recreating images. The proposed method involves creating an algorithm in Python without relying on external libraries. The model will be trained using a dataset of images to learn how to generate pictures

resembling the input data. We will base this model on image reconstruction with a Genetic Algorithm. The Genetic Algorithm, inspired by Charles Darwin's theory of natural evolution, will serve as a foundation for our approach. It imitates natural selection, where individuals with superior traits are selected for reproduction, leading to descendants with enhanced characteristics. The goal is to recreate a degraded or incomplete image using a Genetic Algorithm implemented entirely in Python.

First, using the Genetic Algorithm in Python, the process begins with representing potential solutions as chromosomes. Each candidate image is encoded with a chromosome as a 2D array, where each element represents a pixel value, 0 for black and 1 for white. After initializing a population of candidate random images, each image is evaluated. This step involves comparing the images with target images using pixel-wise difference or structural similarity indexes. This will measure how well both pictures are.

Following this, individuals are selected from the population for reproduction based on their fitness levels. This selection will follow natural selection, favoring individuals with higher fitness scores. Crossover operations are then performed between the selected individuals to produce offspring. This involves exchanging pixel values between parent images to generate new candidate images.

Finally, the iteration through the offspring will continue until a termination condition is met. This termination will include reaching a maximum number of generations and achieving a satisfactory fitness improvement. Using this iterative process, we can reconstruct images using natural selection and evolution principles.

Previous work has been done on the subject. Building upon earlier work in image reconstruction and Genetic Algorithm-based approaches, our proposed method aims to develop a Python program capable of recreating images through an algorithm implemented entirely on Python without reliance on external libraries.

Necessary Materials

We will require several materials to implement the image reconstruction using a Genetic Algorithm in Python, including computational resources, comprehensive documentation, and references.

Firstly, computational resources are essential for executing Python efficiently. Personal computers or laptops are necessary for running the image reconstruction algorithms effectively. Each member should have access to Python on their computer regarding the development environment. We will also use GitHub so both team members can work on the project together.

The team will need to design and implement core Genetic Algorithm operations. Acquiring the necessary materials involves referencing textbooks, academic papers, and online resources to guide implementation in Python. Additionally, throughout testing and validation procedures, using sample images and data is essential to ensure the correctness and effectiveness of the implemented algorithms.

Evaluation

We will evaluate the output of the method by comparing how accurate the image will be to the original image that the AI had been given and if the algorithm made any mistakes. We decide if it worked well or not based on whether the new image looks just like the original or is very similar to the original image. We aim to get the reconstructed image as close to perfect as possible. The AI will not be considered a failure unless it does not stop generating the image, for example, if it continues to keep trying without ending the process.

Along with the evaluation we will do together, we will evaluate each other on how much we collaborate on the project by adding within the code who has done what in the project and how evenly we work on it together.

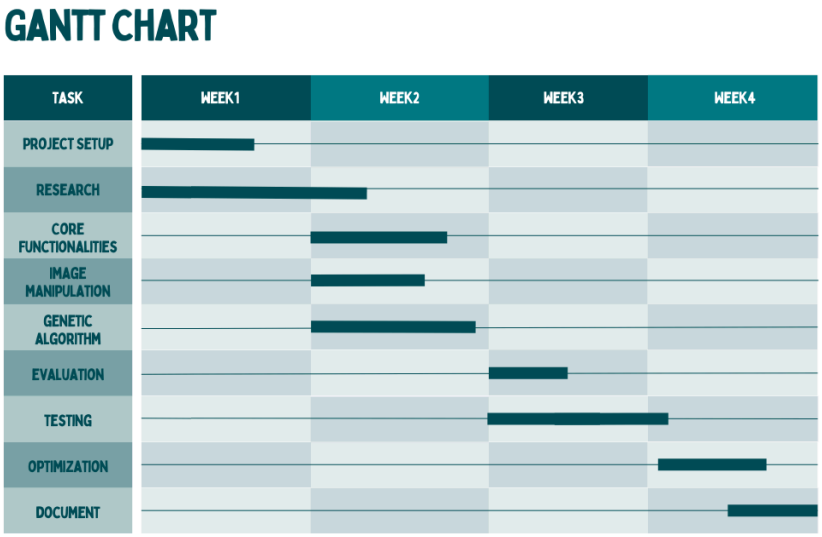
Timeline

Since we will collaborate closely throughout the project, our individual tasks and timelines will largely align, facilitating seamless progress toward our shared goal. In a four-week timeline allocated for the image reconstruction project using a Genetic Algorithm, our approach is to smooth the process, focusing on essential tasks to ensure timely completion. During the first week, we will set up the project, research Genetic algorithm principles and image reconstruction techniques, and assign specific tasks to each team member. We will also prepare a Gantt chart to outline key milestones and dependencies for the project.

Transitioning into the second week, our primary focus will be implementing core functionalities. This phase entails developing essential functions for image representation and manipulation alongside designing and implementing classes and functions fundamental to Genetic Algorithm operations. By integrating these components, we aim to establish a robust framework that facilitates the effective execution of the Genetic Algorithm for image reconstruction tasks.

As we progress into the third week, we will shift our focus towards the crucial tasks of fitness evaluation and testing. Here, the team will try to construct functions to assess candidate images' likeness to the target image. At the same time, we will commence preliminary testing procedures to validate the accuracy and efficacy of our Genetic algorithm implementation.

Finally, in the fourth week, our focus will broaden to encompass optimization, documentation, and finalization. This phase will demand accurate attention to detail as we strive to refine Genetic Algorithm parameters to enhance performance within our limited timeframe's constraints. We will document our code implementation with comprehensive comments, function descriptions, and usage instructions. A final review of the entire project will ensure coherence, completeness, and accuracy across all the work.



Risk Disclosure

Below are some potential risks we will need to consider while creating this project.

Data Availability and Quality: When training the AI model to create an image, we need to ensure that the quality of the data we are giving it is of good enough quality to help the AI, not hinder it. Since this is our first time making an algorithm like this, it will be difficult to determine how complex we can make the images before the algorithm fails if we can even give the AI more than two color images. If we run into this issue, we will go back to simpler images and research different ways to resolve this issue if possible.

Time Constraints: One of the risks we may run into when creating this is the amount of time we have to create it. Although it is a pretty simple project, it is our first time creating a project like this, and we are not very familiar with how these kinds of algorithms are made, so it may take time to research the concept further. To prevent this issue, we will ensure that we stick to the timeline created above and, if we run into issues, seek help if needed.

Out of our Scope: Again, since this is our first time messing around with creating an AI, we could end up having ideas that make the project too difficult to finish within the given time frame. To avoid this issue, we will collaborate with each other as frequently as possible while

working on this, if not working at the same time, to avoid the project becoming bigger and out of our reach to finish.

References

[1]S. Jeswara, “2D image reconstruction using Genetic Algorithm,” *Medium*, Nov. 27, 2021. https://medium.com/analytics-vidhya/2d-image-reconstruction-using-genetic-algorithm-e6ab1c2ea073 (accessed Mar. 25, 2024).

[2]“AI Image Generators for Beginners: A Guide to Get Started,” *Impossible Images*, May 12, 2023. https://impossibleimages.ai/ai-image-generators-for-beginners-a-complete-guide/ (accessed Mar. 25, 2024).

[3]“AI Image Generation Explained: Techniques, Applications, and Limitations,” *AltexSoft*, Jul. 10, 2023. https://www.altexsoft.com/blog/ai-image-generation/ (accessed Mar. 25, 2024).

[4]P. Knoll and S. Mirzaei, “Validation of a parallel genetic algorithm for image reconstruction from projections,” *Journal of Parallel and Distributed Computing*, vol. 63, no. 3, pp. 356–359, Jan. 2003, doi: https://doi.org/10.1016/s0743-7315(03)00019-4 (accessed Mar 25, 2024).

[5]A. Gad, “Reproducing Images using a Genetic Algorithm with Python,” *Medium*, Sep. 23, 2021. https://heartbeat.comet.ml/reproducing-images-using-a-genetic-algorithm-with-python-91fc701ff84 (accessed Mar. 26, 2024).

[6]D. M, “How to use Genetic Algorithm for Image reconstruction?,” *Analytics India Magazine*, Jul. 06, 2022. https://analyticsindiamag.com/image-reconstruction-using-genetic-algorithms/ (accessed Mar. 25, 2024).

[7]A. Piwanska and U. Grycuk, “Images Reconstruction with Use of a Genetic Algorithm,” *IEEE Xplore*, Jul. 16, 2007. https://ieeexplore.ieee.org/document/4273534 (accessed Mar. 26, 2024).

[8]Grammarly, “Write your best with Grammarly.,” *Grammarly.com*, 2023. https://www.grammarly.com

(Grammarly is referenced as it was used for grammar purposes.)