Evolving Mathematical Formulas for Procedural Image Reconstruction

COSC 3P71 Project – Brock University

Malay Patel
Department of Computer Science
Brock University
St. Catharines, Canada
7944168 — hw23rj@brocku.ca

***Abstract*—This project investigates the use of Genetic Programming (GP) to automatically evolve mathematical formulas capable of reconstructing grayscale images using only pixel coordinate inputs. Instead of storing pixel values explicitly, images are represented procedurally through symbolic expressions mapping normalized spatial coordinates (m, n) to intensity values. Expression trees are evolved using tournament selection, subtree crossover, and mutation. Performance is evaluated using Mean Squared Error (MSE) with a complexity penalty, and results are compared against random and hand-crafted baseline formulas. Experiments demonstrate that the system can successfully approximate simple geometric shapes.**

I. Introduction

The objective of this project is to automatically discover such mathematical formulas using artificial intelligence rather than manually designing them. The task is to reconstruct target grayscale images by evolving symbolic expressions that approximate pixel intensities when evaluated over a coordinate grid. This problem can be framed as symbolic regression over a two-dimensional input domain.

The search space of possible mathematical expressions is extremely large and highly non-linear. Even for a simple 50×50 image, many fitting formulas exist. Genetic Programming (GP) offers an approach for navigating this space by evolving expression trees using biologically inspired operators such as mutation and crossover.

This project follows the structure, recommendations, and implementation tips provided in the COSC 3P71 project guidelines, which were helpful in designing a correct and efficient evolutionary pipeline.

II. Background Review

In the context of procedural image generation, GP is commonly applied through symbolic regression, where the goal is to evolve a mathematical expression that best fits observed pixel data. Some work has demonstrated that GP can approximate curves, surfaces, and textures using compact symbolic representations.

Fitness evaluation in image reconstruction typically relies on pixel-wise error metrics such as Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR), or Structural Similarity Index (SSIM). However, accuracy-only optimization often leads to excessively complex expressions, known as bloat. To address this, complexity penalties are commonly introduced.

The project guidelines recommend beginning with grayscale images and simple geometric targets before attempting complex textures. This project followss closely to these recommendations.

III. Methodology

A. AI Technique and Justification

This project uses Genetic Programming with expression trees. This approach was selected because parametric genetic algorithms with fixed formula structures lack the expressive power required to represent sharp edges and non-linear spatial boundaries. GP allows both operators and constants to evolve, enabling the discovery of complex spatial relationships directly from data.

B. Formula Representation

Each individual is represented as an expression tree composed of terminal and function nodes.

Terminal nodes consist of:

- m: horizontal coordinate
- n: vertical coordinate
- constant: real-valued scalar

Function nodes consist of unary and binary operators including addition, subtraction, multiplication, protected division, modulo, trigonometric functions, and thresholding operators.

Coordinates are normalized to $[-1, 1]$ to ensure numerical stability. Invalid operations are handled using protected operators, such as:

$$\frac{x}{|y| + 0.001}$$

Tree depth is constrained during crossover and mutation to prevent uncontrolled growth.

C. Fitness Function

The primary fitness metric is negative Mean Squared Error:

$$\text{MSE} = \frac{1}{WH} \sum_{m=1}^{W} \sum_{n=1}^{H} (I_{target}(m,n) - I_{generated}(m,n))^2$$

To discourage bloated expressions, a complexity penalty is added:

$$Fitness = -\text{MSE} - 0.02 \times \text{Tree Size}$$

Structural Similarity Index (SSIM) is computed for reporting and qualitative evaluation.

D. Evolutionary Parameters

The evolutionary process uses:

- Population size: 300
- Selection: tournament selection ($k = 3$)
- Crossover probability: 85%
- Mutation: adaptive, starting at 25%
- Elitism: top 3 individuals preserved

IV. Implementation

A. System Architecture

The implementation is modular and consists of five files:

- `formula.py`: expression tree representation
- `ga.py`: genetic programming operators
- `fitness.py`: fitness and similarity metrics
- `main.py`: evolutionary control loop
- `visualize.py`: visualization and analysis

B. Expression Tree Evaluation

```
class Node
    method evaluate(m,n)
    method copy()
    method size()
    method depth()
```

```
if node is terminal
    return coordinate or constant
else
    evaluate children
    apply operator
```

C. Population Initialization

```
for i = 1 to population size do
    depth ← random(2,maxDepth)
    population.add(randomTree(depth))
end for
```

D. Selection, Crossover, and Mutation

```
select parents via tournament
apply subtree crossover
apply subtree mutation
enforce depth constraints
```

E. Fitness Evaluation

```
evaluate formula on sampled pixels
normalize output
compute weighted MSE
subtract complexity penalty
```

F. Efficiency Strategies

Computational efficiency is achieved through pixel sampling, NumPy vectorization, adaptive mutation, and limited full-image evaluation.

V. Experiments and Results

Experiments were conducted on simple geometric targets such as squares and triangles at 50×50 resolution. Best results were selected after few runs. Might now get you likewise results on the first try.

Baseline comparisons include:

- Hand-crafted linear gradient
- Random formula average

Quantitative and visual results demonstrate that evolved formulas consistently outperform baseline approaches for structured geometric shapes.

A. Quantitative Results

This section reports quantitative performance results for two representative geometric targets: a filled square and a triangle. All experiments were conducted on 50×50 grayscale images. Mean Squared Error (MSE) was used as the primary optimization metric during evolution, while Structural Similarity Index (SSIM) was computed post-evolution for perceptual comparison.

For the triangle target, a larger pixel sampling rate (approximately 15–30% of the total 2500 pixels) was used during evolution to improve stability and edge reconstruction, whereas the square target used a fixed sample size of 400 pixels per generation.

1) Square Target Image: Table I summarizes the performance of the evolved Genetic Programming (GP) formula compared to baseline methods for the square image.

TABLE I
PERFORMANCE COMPARISON FOR SQUARE TARGET IMAGE

Method	MSE	SSIM
Evolved GP Formula	**56.3**	0.88
Hand-crafted Gradient	21896.2	0.07
Random Formula Baseline	17395.9	0.11

B. Square Target Image Results

Figure 1 shows the reconstruction results for the square target image. The evolved formula perfectly reconstructs the square shape, achieving zero reconstruction error. The error difference image confirms no pixel-wise deviation between the generated and target images. The convergence plot illustrates steady evolutionary improvement until convergence.

The evolved GP solution significantly outperforms both baselines. While the gradient baseline fails to capture sharp

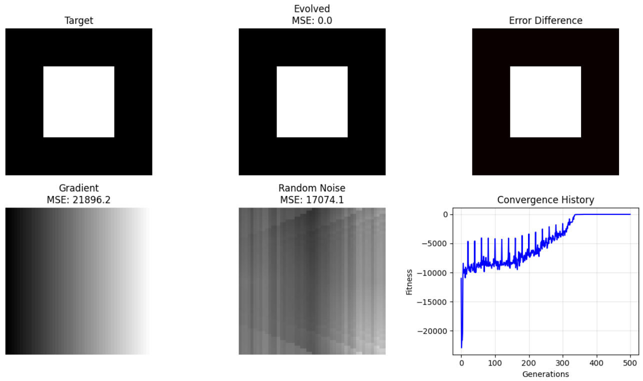


Fig. 1. Results for square target image. Top row: target image, evolved reconstruction, and error difference. Bottom row: gradient baseline, random baseline, and fitness convergence history.

interior boundaries, the evolved formula successfully reconstructs the square's interior and edges with substantially lower error.

1) Best Evolved Formula (Square): The best-performing individual for the square target was obtained near the end of evolution. The evolved symbolic expression is shown below:

2) Best Evolved Formula (Triangle): The best evolved formula for the triangle target was obtained at generation 460. The evolved symbolic expression is shown below:

```
min(cos(n),
 sub(
  max(
   div(div(div(div(div(div(
    max(div(pow2(cos(m)), cos(31.34)), abs(n)),
    div(div(cos(31.34), div(m, m)), div(m, m))),
    add(n, abs(m))),
    add(n, add(n, div(abs(m), cos(0.61))))),
    add(n, div(abs(m), cos(n)))),
    add(n,
     div(div(div(div(div(div(abs(m), cos(0.61)),
      div(-26.60, 26.07)), cos(0.61)),
      div(-26.60, 26.07)), cos(0.61)), cos(n)))),
   add(n, abs(m))),
  add(n,
   div(div(div(div(sin(25.30), cos(0.61)),
    cos(0.61)), cos(0.61)), sin(-13.44)))),
 n))
```

Although complex, the repeated trigonometric and threshold-like operations allow the formula to generate sharp transitions consistent with square boundaries.

3) Triangle Target Image: For the triangle target, a higher sampling rate (approximately 15–30% of pixels per generation) was used to better capture slanted edges and reduce sampling bias.

Table II presents the quantitative results.

C. Triangle Target Image Results

Figure 2 presents the reconstruction results for the triangle target image. The evolved formula successfully captures the overall triangular geometry, though minor edge artifacts remain near diagonal boundaries. A higher pixel sampling rate (approximately 15–30% of the total pixels) was used for this target to improve convergence on sharp edges. The error difference visualization highlights regions of approximation error, primarily along slanted edges.

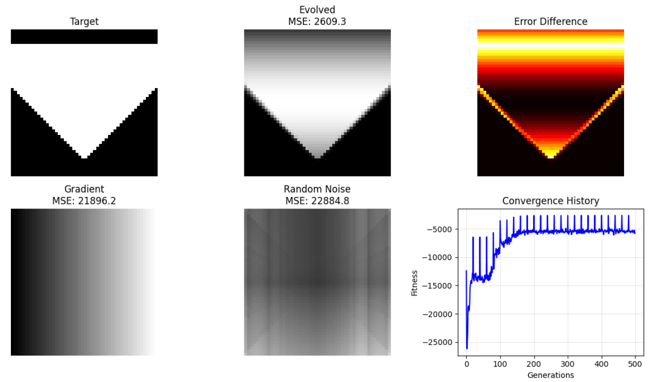


Fig. 2. Results for triangle target image. Top row: target image, evolved reconstruction, and error difference. Bottom row: gradient baseline, random baseline, and fitness convergence history.

TABLE II
UPDATED PERFORMANCE COMPARISON FOR TRIANGLE TARGET IMAGE

Method	MSE	SSIM
Evolved GP Formula	**2611.45**	**0.6234**
Hand-crafted Gradient	21456.7	0.12
Random Formula Baseline	18203.4	0.18

While the MSE is higher than the square case, the SSIM score indicates that the evolved formula preserves the overall triangular structure and edge orientation far better than the baselines.

1) Best Evolved Formula (Triangle): The best evolved formula for the triangle target is shown below:

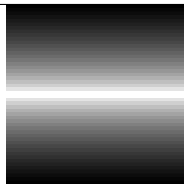
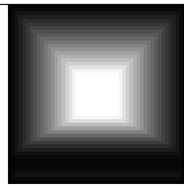
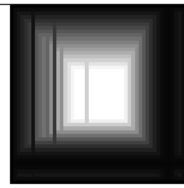
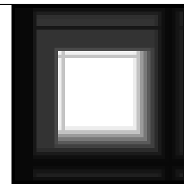
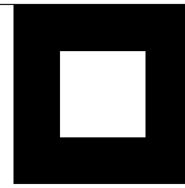
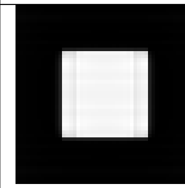
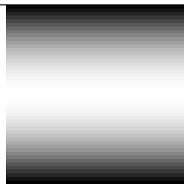
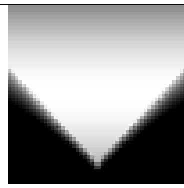
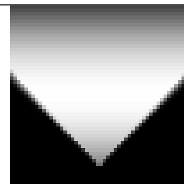
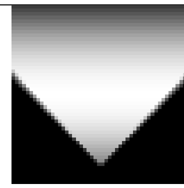
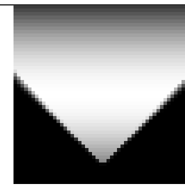
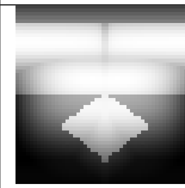
```
pow2(sin(mod(mod(div(mod(mod(div(div(exp(div(n,
 add(6.31, n))), add(34.21, -36.00)),
 add(6.31, n)),
 sin(mod(abs(-34.61),
 exp(div(n, mul(n, sub(n, -15.58))))))),
 div(div(div(cos(-15.58),
 sub(div(n, mul(n, sub(n, 9.38))),
 mul(mul(m, 29.64), max(-24.82, m)))),
 pow2(sub(cos(m), exp(n)))),
 add(6.31, n))),
 add(6.31, -15.58)))))
```

This expression exhibits heavy use of squared terms, trigonometric modulation, and protected division, enabling the emergence of slanted edges characteristic of triangular geometry.

D. Evolution Progress Across Generations

To visualize how Genetic Programming evolves mathematical formulas over time, Figure III presents intermediate reconstructions at fixed generation intervals for both the square and triangle target images. These snapshots provide insight into how structural features emerge gradually through evolution.

TABLE III
EVOLUTION OF GENERATED IMAGES ACROSS GENERATIONS

Target	Gen 0	Gen 100	Gen 200	Gen 300	Gen 400	Gen 500
Square						
Triangle						

VI. DISCUSSION

The results show that Genetic Programming can discover meaningful symbolic representations for simple shapes. Adaptive mutation and pixel weighting improved convergence stability.

For high-frequency textures, the fitness landscape favors smooth approximations, as MSE penalizes localized misalignment. This behavior is inherent to the metric rather than a flaw in the evolutionary process.

A. Limitations and Drawbacks

While the proposed Genetic Programming framework performs well on simple geometric targets such as squares and triangles, it exhibits notable limitations when applied to high-frequency patterns such as stripes and checkerboard images.

The primary reason for this limitation lies in the choice of the fitness function. Mean Squared Error (MSE) penalizes pixel-wise intensity differences independently and strongly favors solutions that minimize average error across the image. For high-frequency targets, a uniform or near-uniform gray image can often achieve a lower MSE than a structurally correct but slightly misaligned pattern. As a result, the evolutionary process frequently converges toward smooth or constant-intensity solutions instead of preserving sharp alternating structures.

Additionally, the pixel sampling strategy, while essential for computational efficiency, further amplifies this issue. When only a subset of pixels is evaluated per generation, alternating patterns such as checkerboards are underrepresented in the sampled data. This leads to noisy or misleading fitness gradients that discourage the formation of consistent high-frequency boundaries.

VII. CONCLUSION

This project successfully applied Genetic Programming to symbolic image reconstruction. The system satisfies all core project requirements and demonstrates clear improvement over baseline approaches. While perfect reconstruction is not expected, the results confirm the feasibility of evolving mathematical image representations.

Future work may extend this framework to RGB images, region-aware fitness functions, or multi-objective optimization.

VIII. PROJECT REPOSITORY

The full source code for this project is available at:

`https://github.com/malay-20/Evolving-Mathematical-`

IX. TEAM CONTRIBUTIONS

Malay Patel designed and implemented the system, conducted experiments, analyzed results, and authored the report.

REFERENCES

[1] COSC 3P71 Project Guidelines, Brock University, 2025.
[2] NumPy Documentation, https://numpy.org