Interactive Evolutionary Art System

Executive Summary:

This report documents the successful enhancement of a TensorGP-based evolutionary art system from a automated, batch-processing tool into a dynamic, interactive web application. The key achievement is the integration of a real-time human-in-the-loop interface, allowing users to guide artistic evolution through visual selection. A significant innovation is the development of a hardware-adaptive version that ensures full functionality even without GPUs or AI models, dramatically increasing the system's accessibility for education and research.

I. PROJECT OVERVIEW & OBJECTIVES

The Project aimed to transform the original emlart_gp.py system (a batch-processing tool for AI-evolved art) into an interactive experience.

The primary Objectives were to:

- Introduce real-time user control via an intuitive web interface
- Develop a hybrid fitness function that combines AI evaluation with human preference.
- Ensure the system remains accessible and functional on hardware without GPUs.
- Provide tools for analyzing and visualizing the evolutionary process.

II. ORIGINAL SYSTEM ARCHITECTURE LIMITATIONS

The original system was functionally robust but offered no interactivity.

- Fully automated: Evolution was driven by AI models (CLIP for text-image similarity and LAION for aesthetic scoring).
- Batch-oriented: Users provided only an initial text prompt and received final outputs, with no ability to steer the creative process.

III. ENHANCED INTERACTIVE SYSTEM: KEY FEATURES

A. Intuitive Web Interface

Built With Gradio, the interface is organized into logic tabs:

- *Setup:* For configuring initial parameters
- *Evolution:* the core interactive tab featuring a real-time gallery of the current population
- *Statistics:* For visualizing performance metrics and evolutionary history.

B. Novel Hybrid Fitnes Evaluation

The system now intelligently blends algorithmic and human-driven assessment:

- *AI Evaluation:* Preserves the original CLIP and

LAION-based scoring for objective metrics.

- *Human Feedback:* User-selected individuals receive a significant fitness boost (a 1.5x multiplier plus a fixed bonus), directly influencing the next generation.
- *Gracefull Degradation:* f AI models are unavailable, the system relies solely on user selection, ensuring uninterrupted operation.

C. Full Evolution History and Navigation

The enhanced system maintains a complete history of the evolutionary run, allowing users to browse through all previous generations sequentially and return to the current generation. Users can view historical populations, see what selections were made in each generation, and analyze the evolution progression through detailed statistics and navigation controls. However, evolution can only continue from the current active generation.

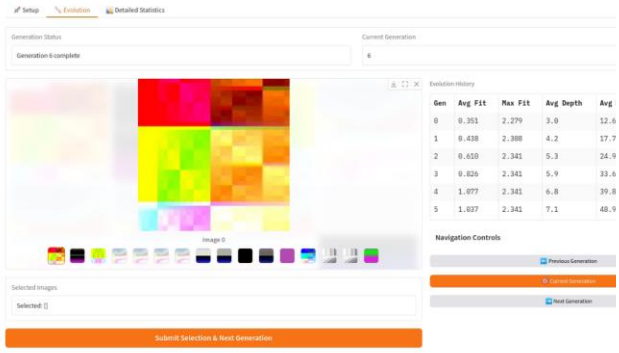

Fig.1 Evolution tab with evolution table history

D. Comprehensive Statistical Dashboard

A dedicated statistics tab provides analytical insights into the evolutionary process through real-time visualization. The dashboard tracks and displays key metrics including fitness progression (average, maximum, and standard deviation), population complexity metrics (tree depth and node counts), and user selection patterns across generations. The system generates comprehensive plots showing fitness evolution over time, complexity trends, and selection pressure patterns. A data table provides generation-by-generation statistics for detailed analysis.

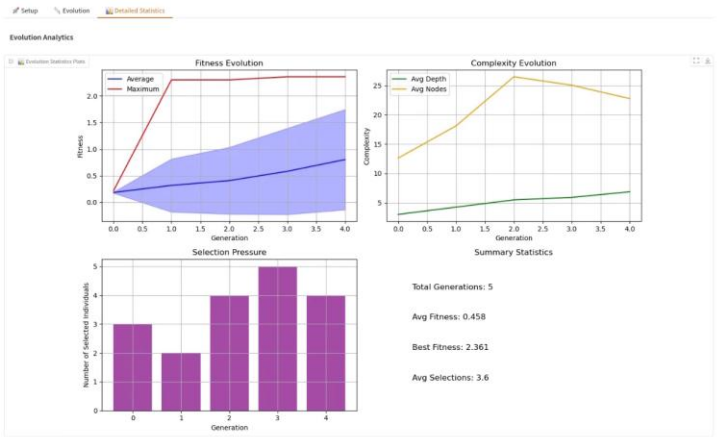

Fig.2 Statistical Dashboard Example

IV. BREAKTHROUGH IN ACCESSIBILITY: GPU-OPTIONAL OPERATION

A major achievement was creating test.py, a version designed for limited hardware.

- Automatic Fallback: The system automatically detects available hardware (CUDA → CPU → Basic Mode).
- Basic Mode: Functions entirely without torch, clip, or laion_aesthetics. User selection becomes the primary driver of evolution.

Benefits: This breakthrough democratizes access by allowing the system to run on standard

laptops, maintains the core educational value of user-guided evolution, and simplifies development and debugging.

V. TECHNICAL IMPLEMENTATION

1. Architecture

The system was built using a modular design, which cleanly separates the powerful TensorGP evolution engine from the user-friendly Gradio web interface. This means each part can be updated, maintained, or even replaced independently, making the entire system more robust and easier to improve in the future.

2. State Management

The application seamlessly manages your entire creative journey. It automatically saves the results of every single generation and remembers all of your selections. This allows you to freely browse through your evolution history, revisit past ideas, and pick up right where you left off without losing any progress.

3. Dependency & installation

For the Full AI-Porwered Experience:

- Core Packages: Python libraries for the interface, math, and image handling (gradio, numpy, PIL).
- AI & Evolution Engine: Pytorch, The CLIP model for understanding prompts, the LAION aesthetic model for "beauty" scoring, and tensorGP framework that drive the evolution.

For the Simple Version (test.py):

- Only need the basic core packages listed above
- It runs entirely without any AI models, making it perfect for learnng the concepts or using any standard laptop.

The system intelligently checks what your computer has available when it starts. If it finds the AI models, it uses them to assist you. If not, it gracefully shifts to a simpler mode where your artistic choices solely guide the evolution, ensuring you can always create.

For detailed, step-by-step installation instructions, please refer to the project's GitHub README: https://github.com/jujuGthb/emlart_project

VI. BENEFITS & IMPACTS

Audience	Key Benefits
Users	Intuitive control, real-time feedback, ability to explore creative possibilities.
Researchers	Studies hybrid evaluation, collects data on human preference, ensures reproducibility.
Educators	Demonstrates evolutionary concepts with no hardware barriers, encourages experimentation.
Developers	Clean, modular codebase with comprehensive error handling and clear documentation.

VII. FUTURE ENHANCEMENT OPPORTUNITIES

- Multi-user Collaboration: Allow groups to guide evolution together.
- Advanced Export: Enable high-resolution image and evolution animation export
- Preference Learning: Train a model to predict user selections to reduce fatigue.
- Cloud Deployment: Package as a scalable web service.

VIII. CONCLUSION

This project successfully transforms evolutionary art from an automated, technical precess into an accessible and engaging platform for human-AI collaboration. The key achievement lies in developing a novel hybrid fitness function that seamlessly integrates objective AI evaluation with subjective human preference, all delivered through an intuitive interface.

The implemented modular architecture provides a robust and extensible foundation, demonstrating an effective model for merging human creativity with algorithmic evolution. This work delivers a functional application that makes the principles of evolutionary computation approachable for a broad audience, from artists to students, opening new possibilities for interactive and personalized art generation.

IX. REFERENCES

- Original Project: https://github.com/cdvetal/emlart-gp-tutorial
- TensorGP Documentation: https://github.com/cdvetal/TensorGP
- Machado, P., Romero, J., & Manaris, B. (2008). Experiments in computational aesthetics: An iterative approach to stylistic change in evolutionary art. The Art of Artificial Evolution, 381-415.
- Gradio Documentation: https://www.gradio.app/docs