Human-Centric Evolutionary Art through Generative Optimization

And its Explainability

Speaker: S. Muhammad Hossein Mousavi

Affiliation: C.E.O. and Senior AI Researcher at Cyrus Intelligence Research

Skills: Emotion Recognition, Image Processing, Data Collection, Optimization, and Synthetic Data Generation.

Contact 1: s.m.hossein.mousavi@cyrusai.ir

Contact 2: s.muhammad.hossein.mousavi@gmail.com

Social: www.linkedin.com/in/smuhammadhosseinmousavi

https://github.com/SeyedMuhammadHosseinMousavi

S. Muhammad Hossein Mousavi

Key Concepts: Foundations for Human-Centric Generative Optimization

Human-Centric Design

Human-centric design means creating processes where algorithms adapt to human emotion, perception, and feedback.

Generative Optimization

This is the fusion of generation (creating new content) and optimization (improving it). Algorithms such as the Genetic Algorithm search for visual or structural solutions that balance beauty, order, and variation, guided by objective functions or human input.

Evolutionary Art

A field where visual art evolves through iterative selection, much like nature. Images are generated, evaluated, and refined over many “generations,” producing artwork that merges algorithmic exploration with aesthetic emergence.

Human-Centered AI: Shin, Youngsoo. "Toward Human-Centered Artificial Intelligence for Users’ Digital Well-Being: Systematic Review, Synthesis, and Future Directions." *JMIR Human Factors* 12.1 (2025): e69533.

Why Human-Centric Generative Optimization?

Problem

Traditional generative systems (neural or algorithmic) often optimize for accuracy or efficiency, but not for aesthetic or human values. They miss the emotional meaning or subjective beauty that humans perceive intuitively.

Need

Human-centric generative optimization connects computational power with human intuition. It treats the human as part of the optimization loop — guiding, evaluating, and shaping outcomes dynamically.

Core Idea

By allowing human feedback to influence algorithmic evolution, AI becomes a co-creator rather than an autonomous generator.

Example

Human feedback can guide a Firefly or Genetic-based optimizer to favor certain visual features, such as warmer color palettes, smoother forms, or more balanced symmetry, embedding aesthetic intent directly into the optimization process.

Generative Optimization: Drefs, Jakob, Enrico Guiraud, and Jörg Lücke. "Evolutionary variational optimization of generative models." Journal of Machine Learning Research 23.21 (2022): 1-51.

Heuristics vs Metaheuristics

Heuristics

- A simple, problem-specific rule or search method that iteratively improves a solution using local information.
- They can easily get stuck in local optima and don't use global exploration strategies.
- Like hill-climbing, greedy search, local beam search, and more.
- One candidate at a time.

Metaheuristics

- A broader framework that guides or controls other heuristics to explore the search space more effectively.
- These include randomness, populations, and other parameters to escape local optima.
- Like Genetic Algorithm (GA), Firefly Algorithm (FA), Differential Evolution (DE) algorithm, Particle Swarm Optimization (PSO) algorithm, and more.
- Population-based.

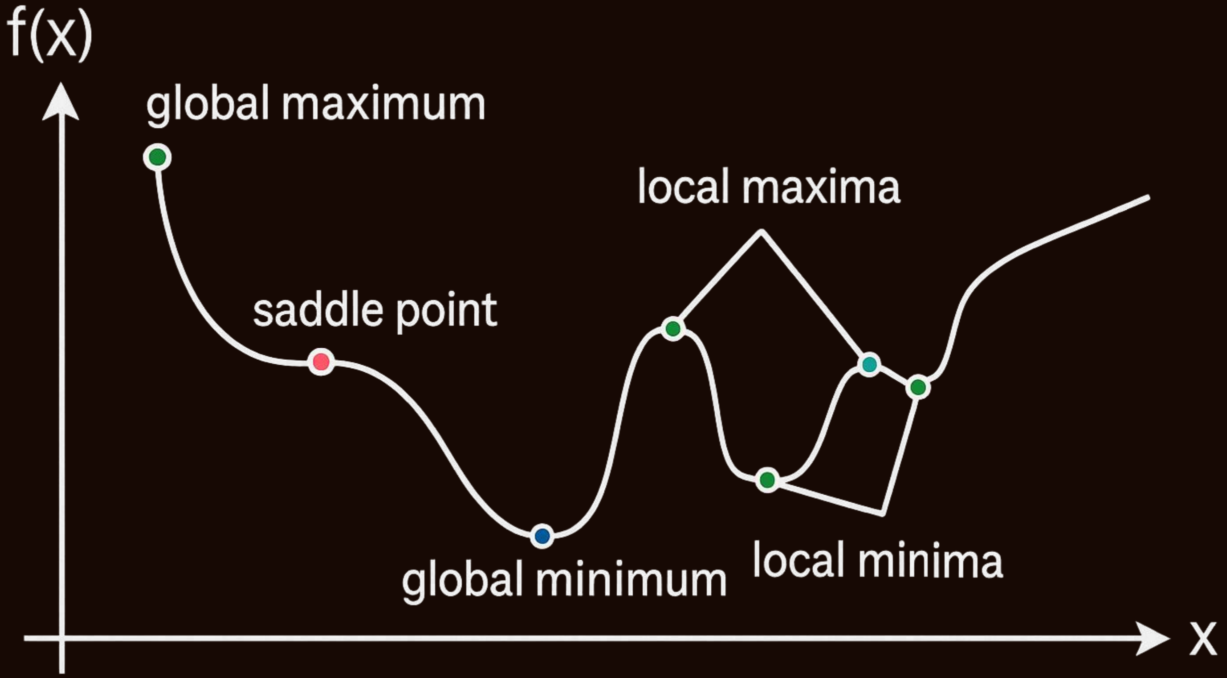
Both operate at the heuristic level, meaning they use experience-based strategies rather than exact mathematical optimization like linear or quadratic programming, but:

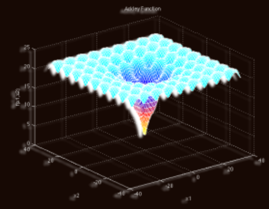
- Heuristics → Generate solutions
- Metaheuristics → Search among solutions (more diverse)

Evolutionary Art: Johnson, Colin G., et al. "Understanding aesthetics and fitness measures in evolutionary art systems." Complexity 2019.1 (2019): 3495962.

optimization landscapes

Global Maximum: The highest value of the objective function across the entire domain of the problem (the best in the case of the fitness function).



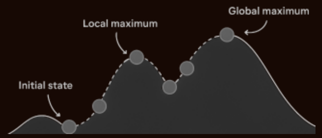


General Optimization: Kochenderfer, Mykel J., and Tim A. Wheeler. Algorithms for optimization. Mit Press, 2019.

Few Algorithms

Hill Climbing

Hill Climbing evaluates neighboring solutions and moves to the one that improves fitness.



Genetic Algorithm

GA is inspired by natural selection, where a population of candidate solutions evolves through selection, crossover, and mutation to find optimal or near-optimal solutions.

Firefly Algorithm

FA simulates the flashing behavior of fireflies, where each firefly is attracted to brighter (better) solutions based on their light intensity, guiding the population toward global optima.

Particle Swarm Optimization

PSO is inspired by the social behavior of birds or fish, where particles move through the search space by following both their own best-known position and the swarm's global best position to find optimal solutions.

Hill Climbing: Burke, Edmund K., and Yuri Bykov. "The late acceptance hill-climbing heuristic." European Journal of Operational Research 258.1 (2017): 70-78.
GA: Holland, John H. "Genetic algorithms." Scientific american 267.1 (1992): 66-73.
FA: Yang, Xin-She. "Firefly algorithms for multimodal optimization." International symposium on stochastic algorithms. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009.
PSO: Kennedy, James, and Russell Eberhart. "Particle swarm optimization." Proceedings of ICNN'95-international conference on neural networks. Vol. 4. ieee, 1995.

From Automation to Co-Creation

1

Traditional optimizers focus purely on numerical fitness — minimizing MSE, maximizing accuracy, or improving convergence.

2

However, art, design, and creativity require subjective judgment that no metric fully captures.

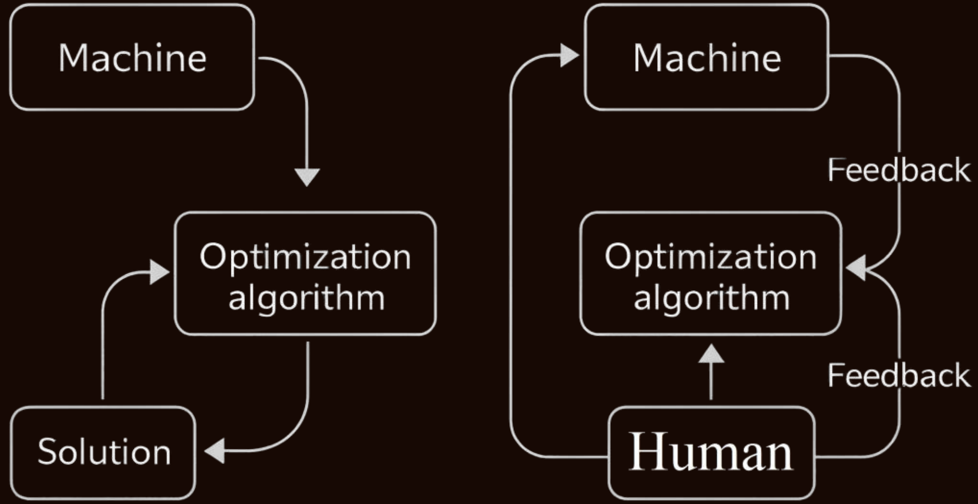
3

Human-in-the-loop optimization allows artists or designers to lead algorithms through aesthetic feedback.

4

This transforms optimization from a mechanical search into an interactive creative dialogue between human intent and computational exploration.

From Automation to Co-Creation



Autonomous optimization

Human-guided optimization

quantitative evaluation from the fitness function

MSE, PSNR, SSIM — numerical scores of how close the image is to the target.

subjective, emotional, or aesthetic judgment

make the color warmer, change shape, opacity, size, and more

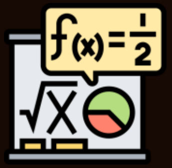
System Architecture of Human-Guided Generative Optimization

1. Input / Target Setup:
   *. A target image or abstract goal is loaded.
   *. The system initializes a blank canvas.
2. Optimization Engine:
   *. Uses a population-based or iterative search (e.g., PSO, GA).
   *. Generates visual candidates or shapes.
3. Machine Evaluation:
   *. Each candidate is scored using quantitative metrics (MSE, SSIM, PSNR).
4. Human Feedback Loop:
   *. User rates or modifies aesthetics (color, shape, opacity).
   *. Feedback biases the next generation of images.
5. Final Composition:
   *. The algorithm merges both human and computational feedback.
   *. Final art.

Optimization and Feedback Dynamics

1. Initial Phase:
   *. Random shapes or patterns are generated (rectangles, circles, triangles, lines).
   *. The system evaluates them with MSE, SSIM, and PSNR.
2. Optimization Loop:
   *. The algorithm adjusts parameters (position, color, rotation, opacity) to minimize error.
   *. In each iteration, fitness improves/error decreases.
3. Feedback Injection:
   *. At fixed intervals, human feedback modifies the search direction.
   *. Example: “Increase warmth,” “add symmetry,” or “reduce opacity.”
   *. The optimizer blends human feedback with numeric fitness ($\alpha \times \text{error} + (1 - \alpha) \times \text{feedback}$).
4. Convergence:
   *. The system stabilizes as both aesthetic (subjective) and computational (objective) goals align.

Feedback-Fitness Combination



*. Feedback-Fitness Combination Formula

$$E' = \alpha \times E_{\text{objective}} + (1-\alpha) \times \left(1 - S_{\text{human}}\right)$$

Where:

- E': new effective error or fitness value
- $E_{\text{objective}}$: objective error (e.g., MSE)
- S_{human}: human feedback score (0–1)
- α: weight balancing between machine accuracy and human preference (e.g., 0.85)

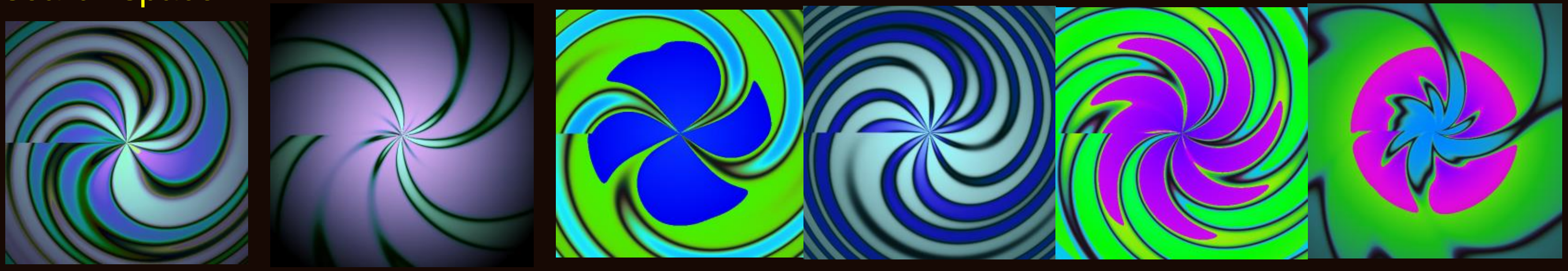
Interpretation:

*. The algorithm reduces MSE but also listens to subjective human judgment.

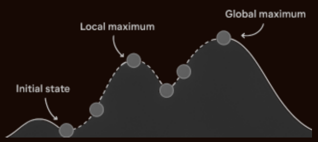
*. So, even if the image is not pixel-perfect, **if the human likes it, it gets rewarded**.

Some Algorithmic Results
Firefly Algorithm

1. It is a Firefly Algorithm–based generative art program, specifically a Firefly Aesthetic Generator.
2. It evolves abstract visual patterns by optimizing aesthetic metrics (symmetry, color balance, composition, color variance, etc.) using the Firefly Algorithm.
3. It creates new artworks that maximize aesthetic quality through iterative optimization.
4. The fitness function evaluates how well an image meets aesthetic principles of harmony, structure, and variety.
5. The spiral colors visualize how fireflies cluster toward the most visually attractive center of the search space.



Some Algorithmic Results
Hill Climbing – the Boat



Start

End

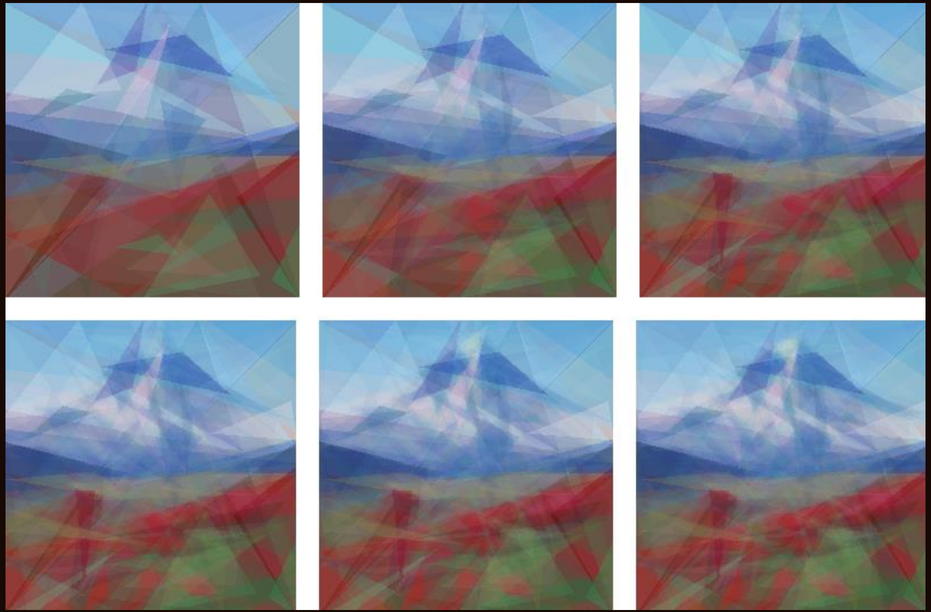
Elements:
1. Rectangle

Target

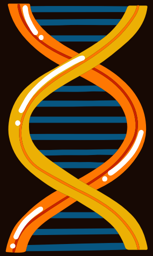
the Boat

Some Algorithmic Results
Genetic Algorithm – Damavand Mountain

Start



End



Elements:
1. Triangle

Target

Damavand Mountain

Some Algorithmic Results
Firefly Algorithm – Cyrus

Start

End

Elements:

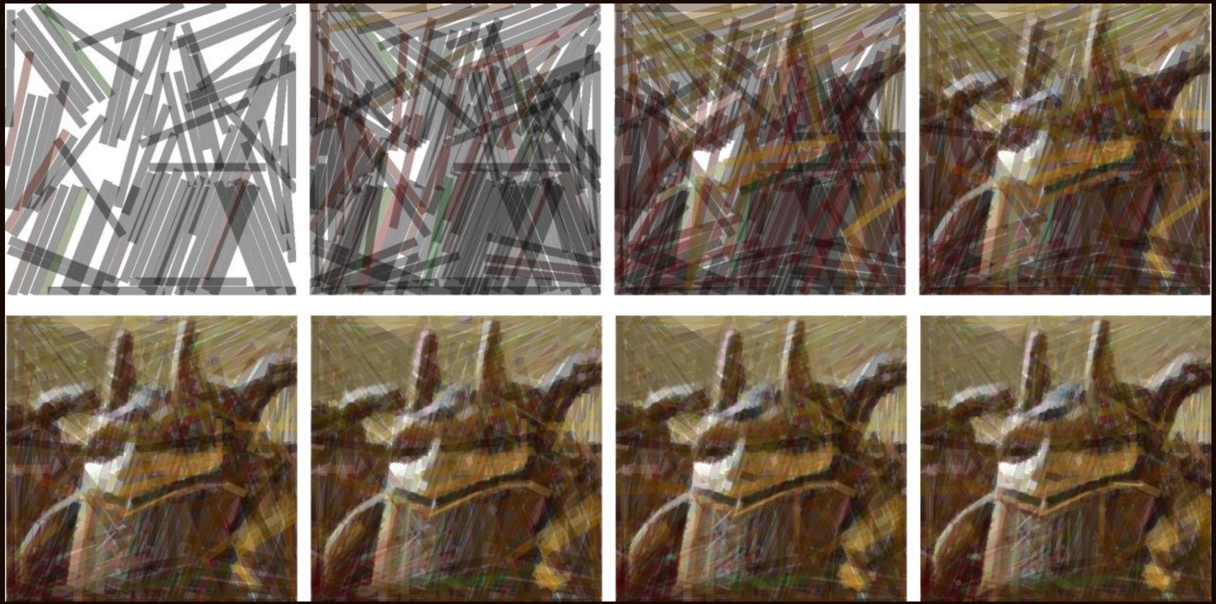
1. Polygon (3 to 7 edges)

Target

Cyrus

Some Algorithmic Results
PSO - Helmet

Start



End

Elements:
1. Line

Target

Helmet

Human-Centric Evolutionary Art through Generative Optimization And its Explainability

Some Algorithmic Results
PSO - Leon

Start

End

Elements:
1. Line
2. Ellipse
3. Triangle

Target

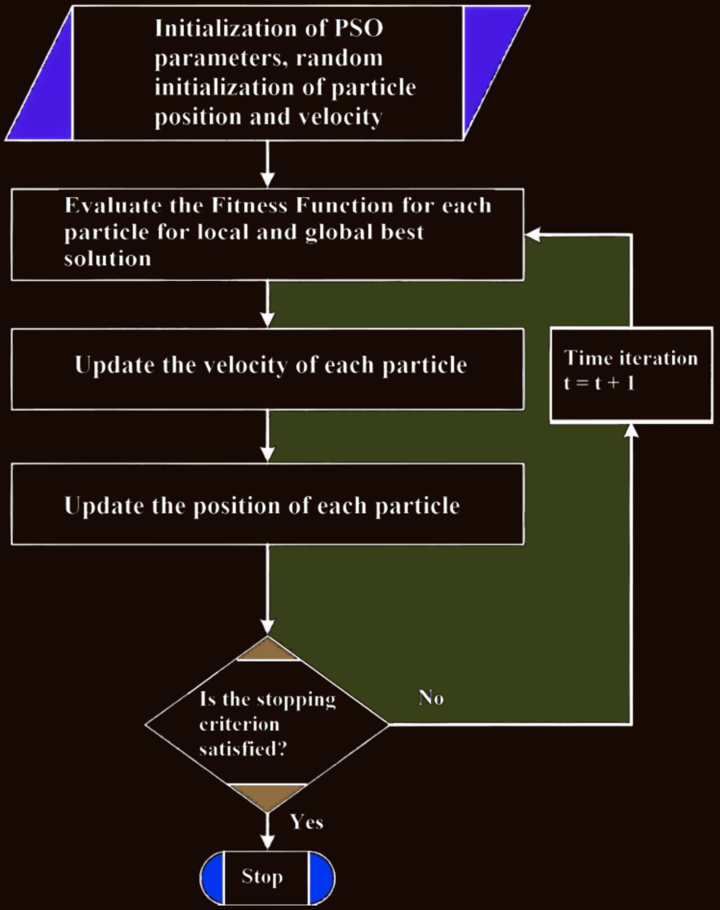
Leon

Human-Guided Multi-Shape PSO Art:
Combining Optimization and Aesthetic Feedback

Human-Guided Multi-Shape PSO Art

PSO is inspired by the social behavior of birds or fish, where particles move through the search space by following both their own best-known position and the swarm's global best position to find optimal solutions.

Particle Swarm Optimization (PSO): Simulates the social behavior of swarms to explore the solution space.



Human-Guided Multi-Shape PSO Art

*. Each particle in PSO represents one possible shape, contributing to the whole image.

*. After deciding on the shape, we have the following parameters for that specific shape:

*. For instance, when we feed back and say rectangle, we have:

>. (x, y): Shape center on the canvas

>. (w, h): Width and height (size)

>. Angle (θ): Rotation

>. (R, G, B): Color channels

>. α: Opacity (transparency)

>. So, each particle = [x, y, w, h, θ, R, G, B, α].

*. The search space is a continuous 9-dimensional space.

*. Each particle tries to find parameters that reduce the difference (MSE) between the rendered canvas and the target image.

Mapping Classical PSO → PSO-Art System

PSO Concept	Mathematical Role	Equivalent
Particle position (x_i)	Candidate solution in search space	Shape parameters [x, y, w, h, θ, R, G, B, α] (where to draw and how it looks)
Particle velocity (v_i)	Direction & speed of movement in search space	How quickly the shape's features (size, color, position) change between iterations
Personal best (pbest)	Best solution found by each particle	The particle's best version of the shape (that produced lowest error with target)
Global best (gbest)	Best among all particles	The single most accurate shape for the current step, added to the final canvas
Fitness function f(x)	Evaluates quality of a solution	Mean Squared Error (MSE) between rendered canvas and target image
Swarm	All particles evolving together	All candidate shapes competing/cooperating to paint one optimal brushstroke.
Iterations	Repeated updates to improve positions	Each PSO loop refining shape geometry, color, opacity until convergence.

How PSO Evolves Shapes to Create the Final Artwork

*. Particle Evolution:

›. Each particle adjusts its parameters (position, size, color, and opacity) through PSO's velocity and position updates.

›. It learns from its own best result (pbest) and the swarm's best (gbest).

*. Iterative Composition. For every shape:

›. PSO finds the best version of that shape type.

›. The best (global) shape is rendered on the canvas.

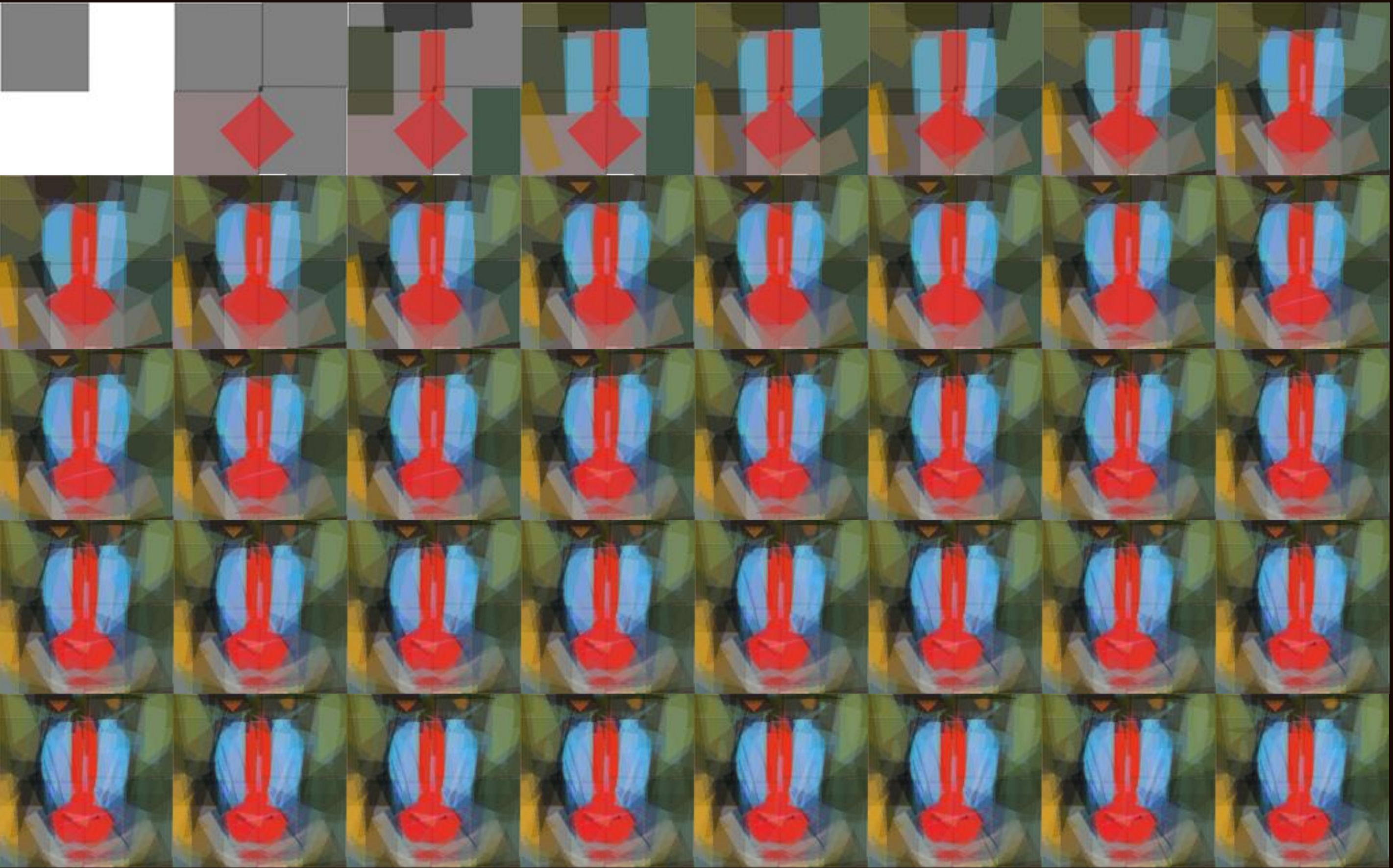
›. Canvas updates, and PSO restarts for the next shape. Gradually, the image accumulates.

*. Evaluation & Feedback:

›. Fitness = Mean Squared Error (MSE) between canvas and target.

›. Human feedback modifies PSO's direction, favoring certain shapes, colors, or opacity levels for aesthetic improvement.

Human-Guided Multi-Shape PSO Art



Shape: Rectangle
Itr: 30
Human rate: 0.4
Human change: shape to a triangle with a 0.5 ratio

Shape: Triangle
Itr: 60
Human rate: 0.5
Human change: color to warmer with a 0.6 ratio

Shape: Triangle
Itr: 90
Human rate: 0.6
Human change: opacity to 0.5 ratio

Shape: Circle
Itr: 120
Human rate: 0.7
Human change: shape to a circle with a 0.8 ratio

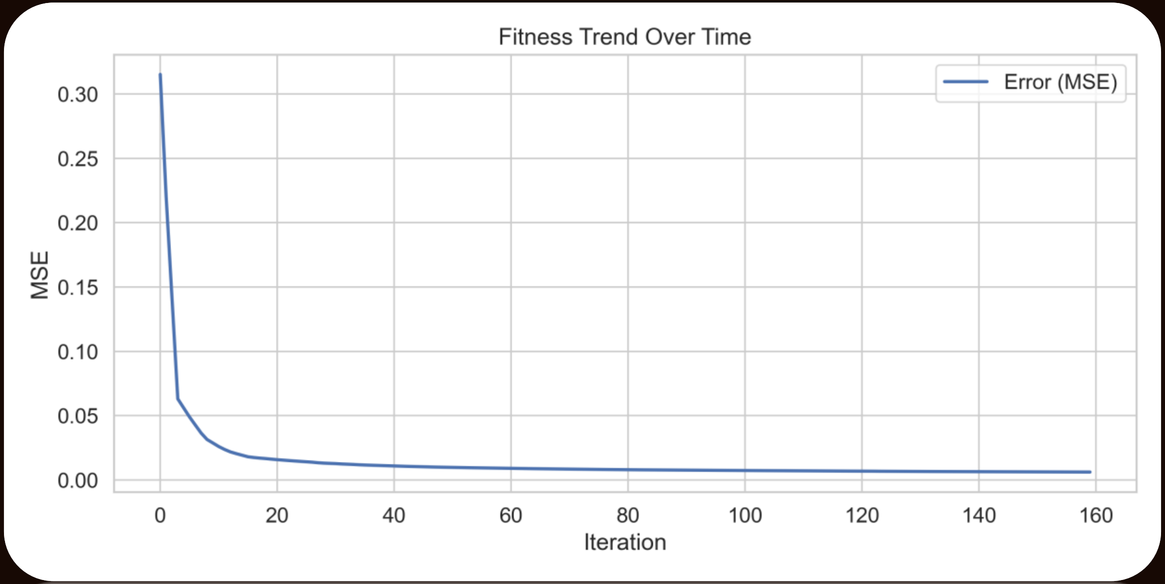
Baboon

MSE: 0.012 to 0.006
PSNR: 19 to 22
SSIM: 0.30 to 0.50

Target

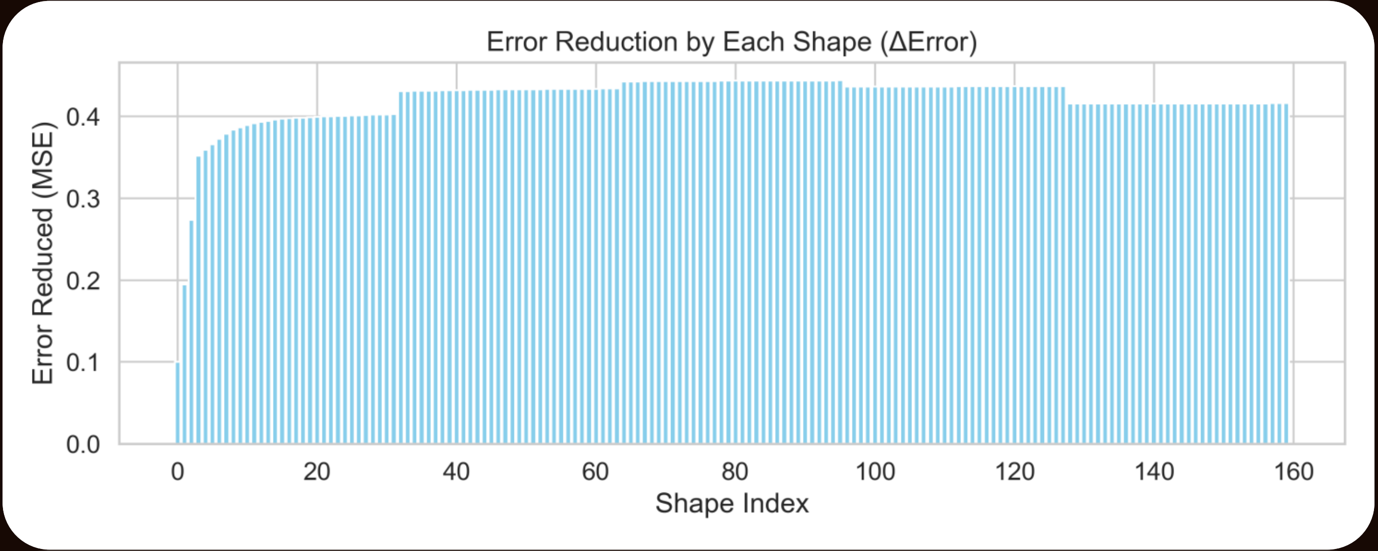
Explainability - Basic

*.Explainability reveals why an AI makes each decision.
*.Explainable Artificial Intelligence (XAI) turns complex models into understandable elements for humans.
*.Here, fitness shows how the base image is reconstructing the baboon image.
*.Fast convergence means the target image foundation is made in the early iterations.



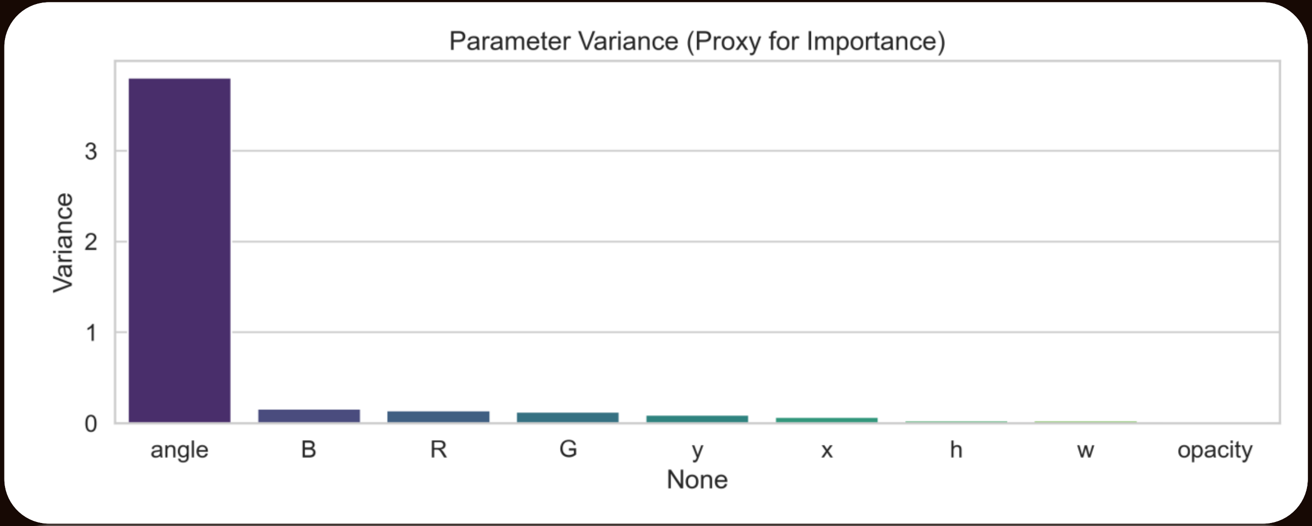
Explainability - Basic

*.Each bar shows how much a single shape reduced the total MSE.
*.Early rectangles caused steep error drops, triangles refined local textures, and circles provided smooth blending.
*.ΔError (delta error) shows the change in MSE after adding each shape
*.It tells how much that specific shape improved the image.
*.A higher error bar means each shape contributed positively to reducing the total error.



Explainability - Basic

*.Angle has by far the highest variance, meaning it changes the most across all shapes and has the strongest influence on reconstruction.
*.Color channels (R, G, B) and position (x, y) vary only slightly, so they fine-tune rather than dominate.
*.Width, height, and opacity barely vary, so they contribute little to optimization.
*.The algorithm relies mostly on rotational adjustment (angle) to fit the shapes, while color and size remain relatively stable.



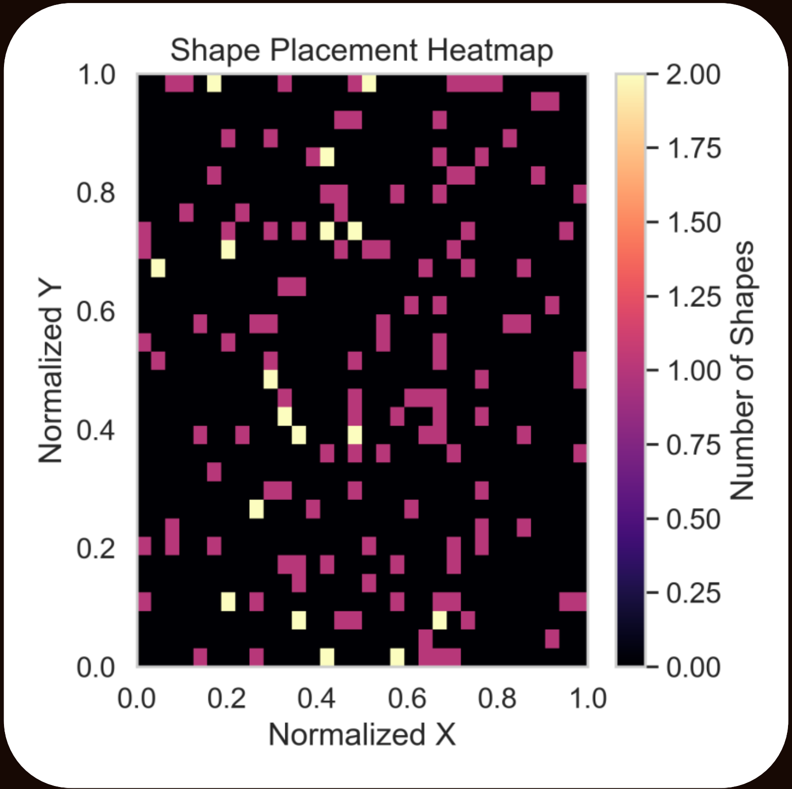
Explainability - Basic

*.Each colored square represents a region of the image canvas.

*.Brighter areas (yellow–white) indicate a higher density of shapes, meaning the algorithm focused more there to reduce reconstruction error.

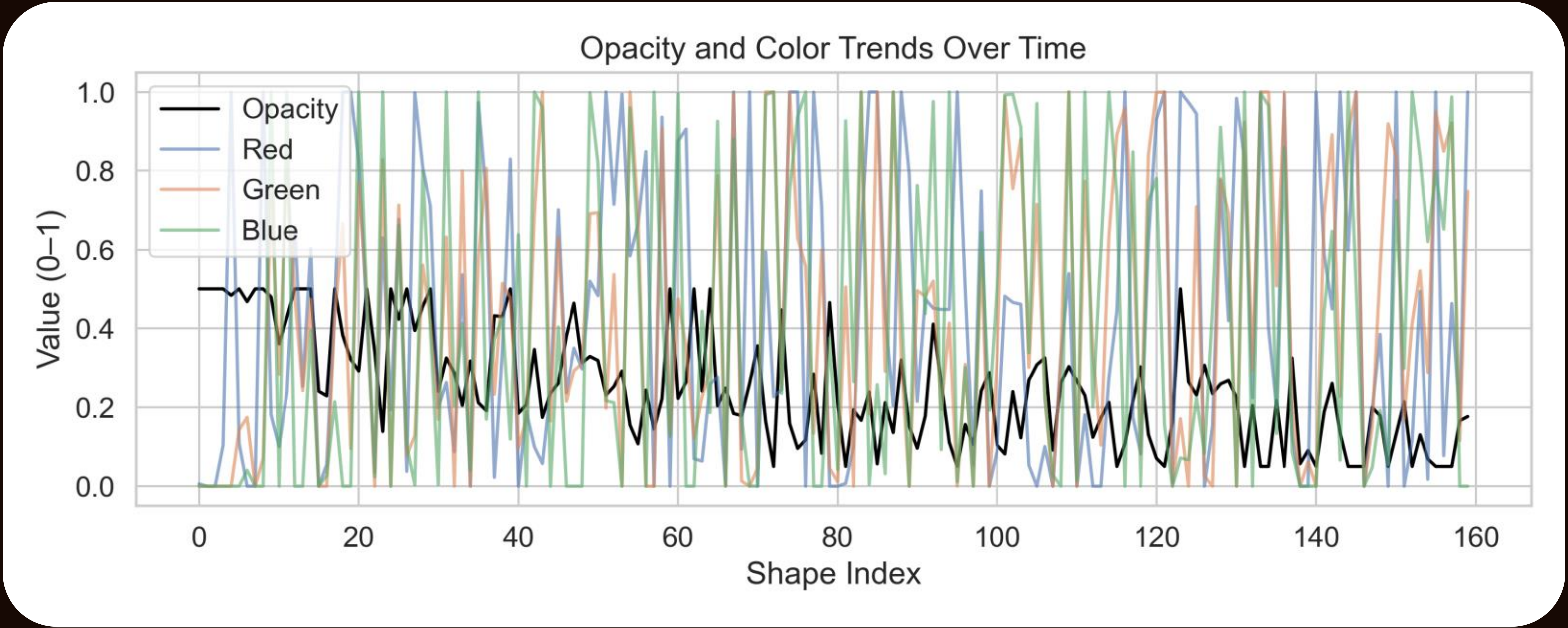
*.Darker regions were less frequently updated, showing those areas either already matched the target early or contributed little to further improvement.

*.So, it shows which parts of the image the PSO-based process found most important for matching the target.



Explainability - Basic

*.The plot shows the evolution of color and opacity for each generated shape throughout the reconstruction.
*.The black line (opacity) shows how transparent each shape was.
*.The R, G, and B curves fluctuate strongly, showing that the optimizer continuously explored different color combinations to best approximate the target texture and tones.



Explainability - Advanced

*. Permutation explainability checks how important each feature really is by seeing how much the model's accuracy drops when that feature's values are mixed up.

*.We have 9 features in total: [x, y, w, h, angle, R, G, B, opacity].

*.In permutation explainability, the algorithm tests each feature one by one.

*.It randomly shuffles (mixes up) that feature's values across all samples, then measures how much the model's accuracy drops.

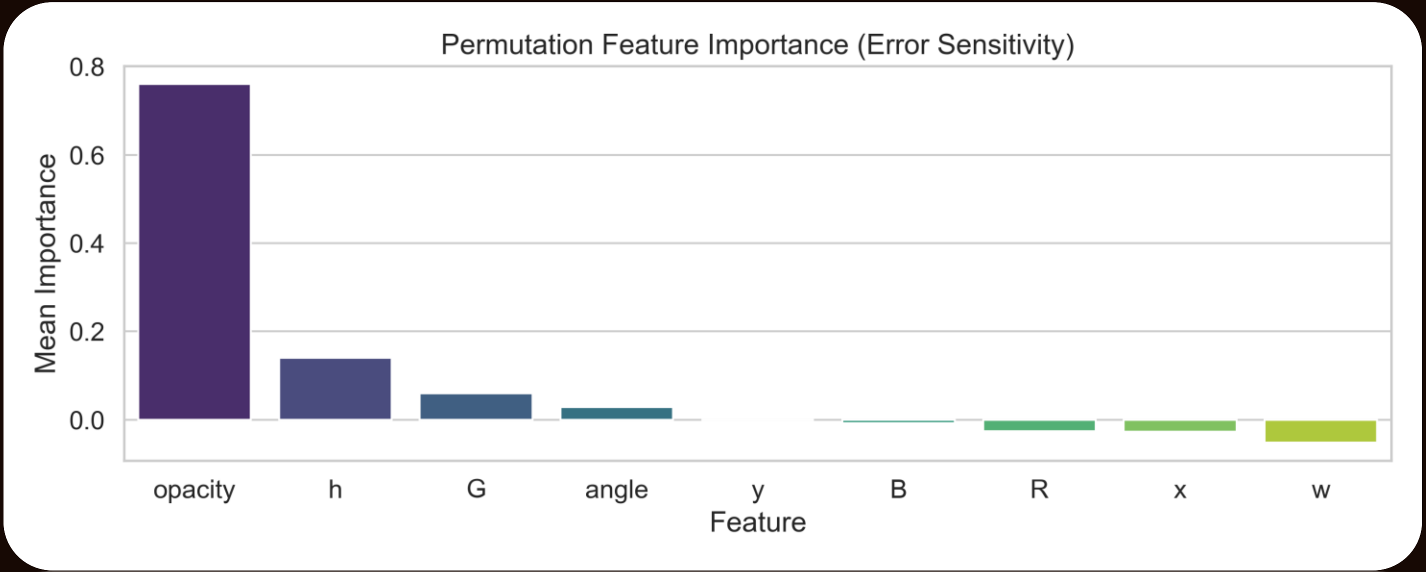
*.This process is repeated many times (here, 20 times) for reliability, and the average performance drop is reported as the feature's importance score.

Explainability - Advanced

*.Opacity dominates as the most influential parameter.

*.Small changes in transparency have the greatest effect on reconstruction accuracy.

*.Height (h) and green (G) channels contribute modestly, while spatial and color coordinates show minimal influence.



Explainability - Advanced

*.LIME (Local Interpretable Model-agnostic Explanations) explains why the model made a specific prediction for one sample.

*.It slightly changes that sample's feature values and observes how the output shifts, building a simple local model (like a small linear one) to approximate the behavior.

*.In short, LIME shows which features mattered most for that one shape or instance, not globally for the whole model.

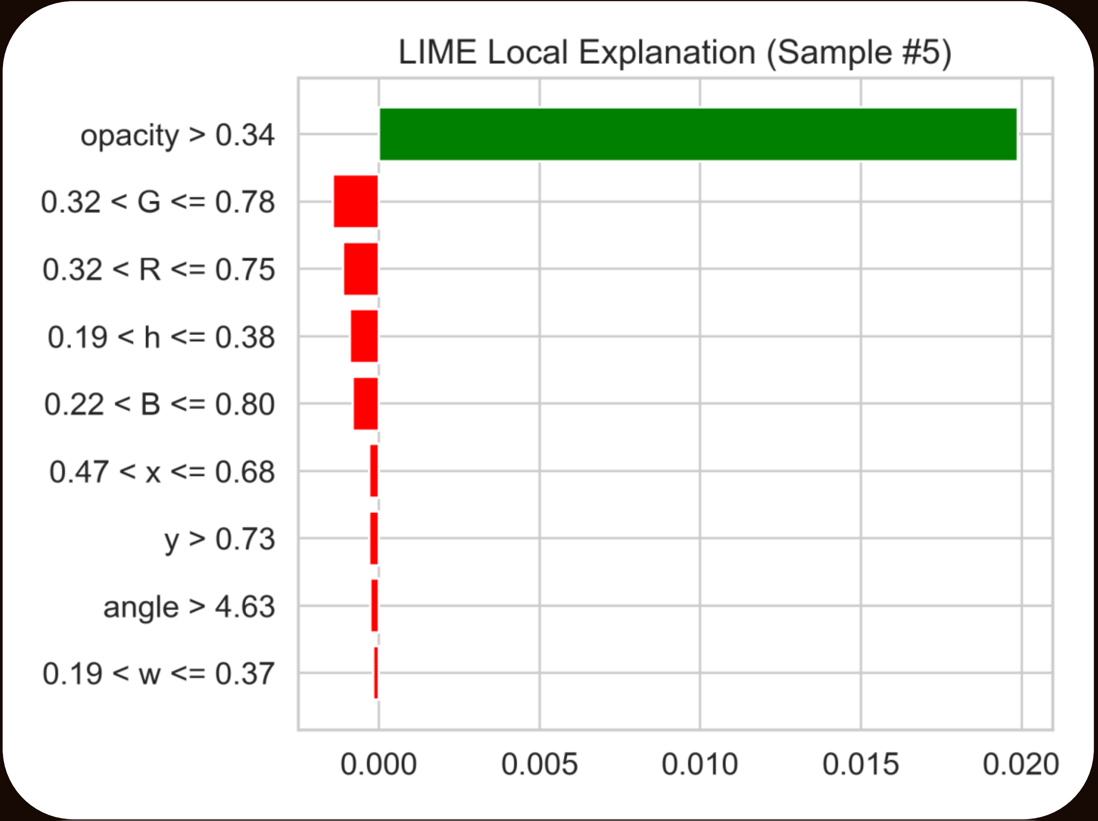
>-Permutation importance → checks global influence by shuffling one feature across all samples and seeing how much model performance drops.
It tells you which features matter overall.

>-LIME → looks at one single prediction and slightly changes the input locally (adds small random changes) to learn which features most affected that one result.

>-So permutation explains the whole model's logic, while LIME explains one specific decision.

Explainability - Advanced

*.The LIME plot shows how individual features affected one sample's prediction.
*.Here, opacity strongly increased the model's confidence while other features like color (R, G, B) and geometry (h, w, angle) had smaller or negative effects.
*.In short, it means the local decision for this shape was driven mainly by its opacity value.
*.The sample 5 here is the 6th rectangle inside the test set.



Explainability - Advanced

*.Both Permutation Importance and LIME identify opacity as the dominant factor.

*.That means the model is consistent: opacity strongly influences both the global behavior (across all shapes) and the local prediction (for that one sample).

THANKS!

Do you have any questions?

s.m.hossein.mousavi@cyrusai.ir
s.muhammad.hossein.mousavi@gmail.com
https://cyrusai.ir/

It's been a pleasure sharing this journey of optimization and explainability.

Human-Centric Evolutionary Art through Generative Optimization and its explainability.
Presented by Seyed Muhammad Hossein Mousavi