

# Compte Rendu de Projet – Génération d'image par Algorithmes Génétiques

Marianne B. – Double Majeure Informatique

12 juin 2025

## 1 Introduction

Ce projet a pour objectif de générer automatiquement une image imitant une image cible à l'aide d'algorithmes génétiques. L'image est reconstituée par un assemblage de traits de pinceau simulés (brush strokes), selon une logique d'évolution inspirée de la biologie.

## 2 Contexte et Objectifs

L'idée est de commencer avec une population d'images générées aléatoirement, chacune composée de formes primitives (triangles, ellipses, traits, etc.). Ces formes évoluent au fil des générations pour ressembler de plus en plus à une image cible. Cela nous permet d'appliquer des concepts de sélection, mutation, croisement et évaluation de fitness à un problème visuel.

## 3 Conception Algorithmique

### 3.1 Représentation des individus

Chaque image est représentée comme un ensemble de "brushs" décrits par :

- Position  $(x, y)$
- Taille et rotation
- Couleur  $(r, g, b)$  et opacité
- ID de brush utilisé dans la bibliothèque

### 3.2 Évolution

Nous utilisons une boucle d'évolution avec :

- Sélection des meilleurs individus (selNSGA2)
- Croisement (crossover uniforme)
- Mutation (paramètres légèrement modifiés)
- Fitness calculée par distance perceptuelle (LAB + deltaE)
- Recherche de nouveauté (HOG + distance entre individus)

### 3.3 Améliorations guidées

- K-means sur les bords pour mieux positionner les brushes
- Gradient de Sobel pour guider la direction des traits
- Évolution par phases (coarse-to-fine)

## 4 Implémentation Technique

Le projet est structuré autour de trois modules Python :

- `core/brush.py` : classe `BrushStroke`, encodage génétique
- `core/painting.py` : rendu de l'image, détection de bords, génération de GIF
- `core/solver.py` : algorithme d'évolution avec DEAP

Un notebook Jupyter permet d'exécuter les tests et d'afficher les résultats facilement.

## 5 Expérimentations

Nous avons testé le système sur plusieurs images (Mario, arbre, etc.). Voici un exemple de résultat après 100 générations :

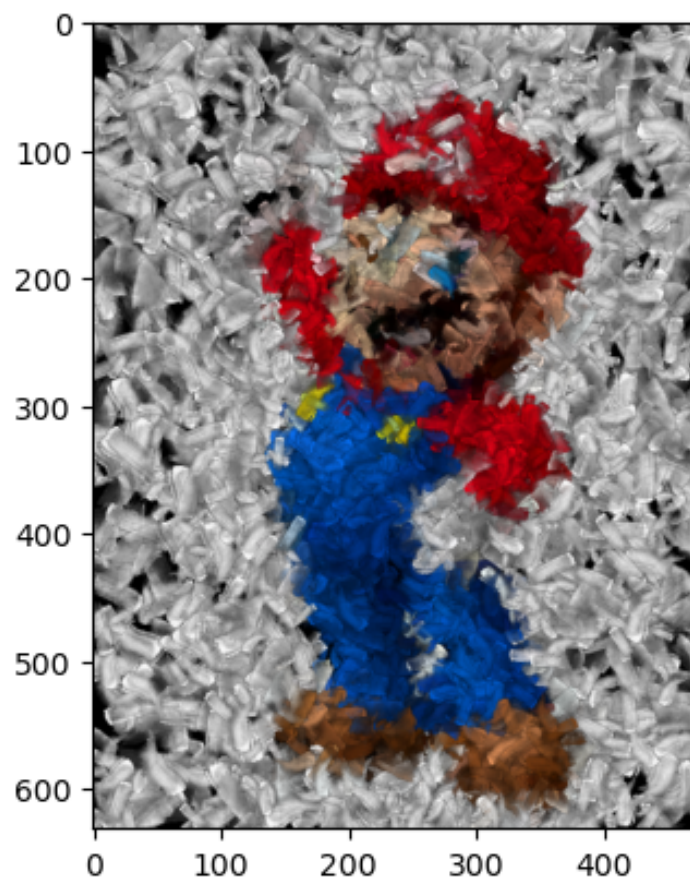


FIGURE 1 – Image générée à partir de Mario après 100 générations

Nous avons constaté que :

- L'algorithme converge rapidement sur les images simples
- Le gradient améliore l'alignement des traits
- La recherche de nouveauté évite la stagnation

## 6 Conclusion

Ce projet nous a permis d'appliquer des techniques avancées d'optimisation évolutionnaire à un problème visuel original. Le code reste modulaire et pourrait être amélioré par :

- L'ajout de textures
- L'intégration de réseaux de neurones
- L'utilisation de masques manuels

**Date de rendu du rapport :** 26 juin 2025 à 23h59

**Durée de la soutenance :** 20 minutes de présentation, 5 minutes de questions