

Generador de Fotomosaicos en Python

Edgar Montiel Ledesma

18 de noviembre de 2024

Introducción

Este programa genera un fotomosaico a partir de una imagen base utilizando pequeñas imágenes de un directorio. Para optimizar la selección de las imágenes más adecuadas, se implementan dos métodos: el método de **Riemersma** y el método de **distancia euclidiana**. Este programa también incluye una interfaz gráfica para facilitar su uso.

Características del Programa

1. Generación de Fotomosaicos

El programa divide la imagen base en pequeñas regiones (*tiles*) y calcula el color promedio de cada región. Luego selecciona imágenes del directorio que coincidan mejor con ese color promedio. El resultado es un mosaico visualmente similar a la imagen original, pero compuesto por las imágenes del directorio.

2. Métodos de Selección de Imágenes

- **Riemersma**: Calcula la distancia euclidiana entre el color promedio de una región y el color promedio de cada imagen disponible, seleccionando la imagen con la menor distancia.
- **Euclidiana**: Método similar al anterior, pero estructurado para evaluar su efectividad en comparación con Riemersma.

```
edgar@edgar-Latitude-3380:~/FCiencias/MateriasOptativas/ProcesoDigitaldeImagen/ProyectoFinal$ python3 FotoMosaico.py
Preprocesando imágenes, esto puede tardar un poco...
Procesadas 63162 imágenes.
Mosaico guardado en /home/edgar/FCiencias/MateriasOptativas/ProcesoDigitaldeImagen/ProyectoFinal/FM1.jpg.
```

Figura 1: Imagen del Proceso.

3. Interfaz Gráfica

Se implementa una interfaz gráfica en Python que permite:

- Cargar la imagen base desde el sistema de archivos.

- Procesar la imagen base para generar el fotomosaico utilizando el método seleccionado.
- Guardar el resultado como un archivo de imagen.
- Elegir entre los métodos de selección de imágenes mencionados.

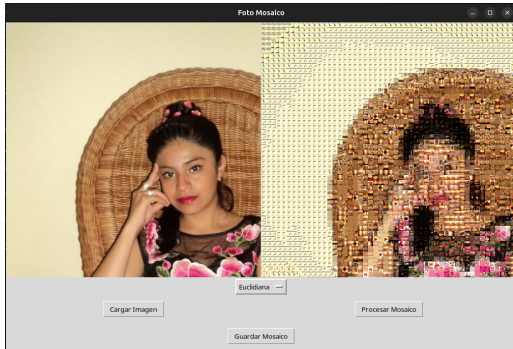


Figura 2: Euclidiana

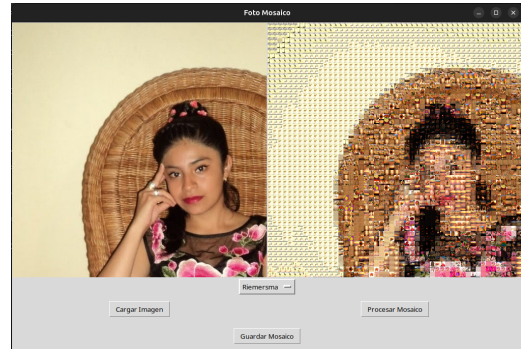


Figura 3: Riemersma

4. Ajuste de Definición

El tamaño de los *tiles* se establece en 20×20 píxeles, lo que incrementa la definición del fotomosaico al permitir más piezas pequeñas.

Detalles Técnicos

- **Librerías utilizadas:**
 - PIL (Pillow) para la manipulación de imágenes.
 - Tkinter para la interfaz gráfica.
- **Ruta de imágenes:** Se espera que las imágenes estén en un directorio llamado `photos-800000`.
- **Preprocesamiento:** El programa calcula el color promedio de cada imagen en el directorio para optimizar el proceso de selección.

Funcionamiento del Programa

1. **Cargar la imagen base:** El usuario selecciona una imagen para procesar.
2. **Seleccionar método:** El usuario elige entre los métodos *Riemersma* o *Euclidiana*.
3. **Generar mosaico:** El programa divide la imagen en regiones, calcula los colores promedio y selecciona las imágenes correspondientes.
4. **Guardar resultado:** El usuario puede guardar el fotomosaico en un formato de imagen (`.jpg` o `.png`).

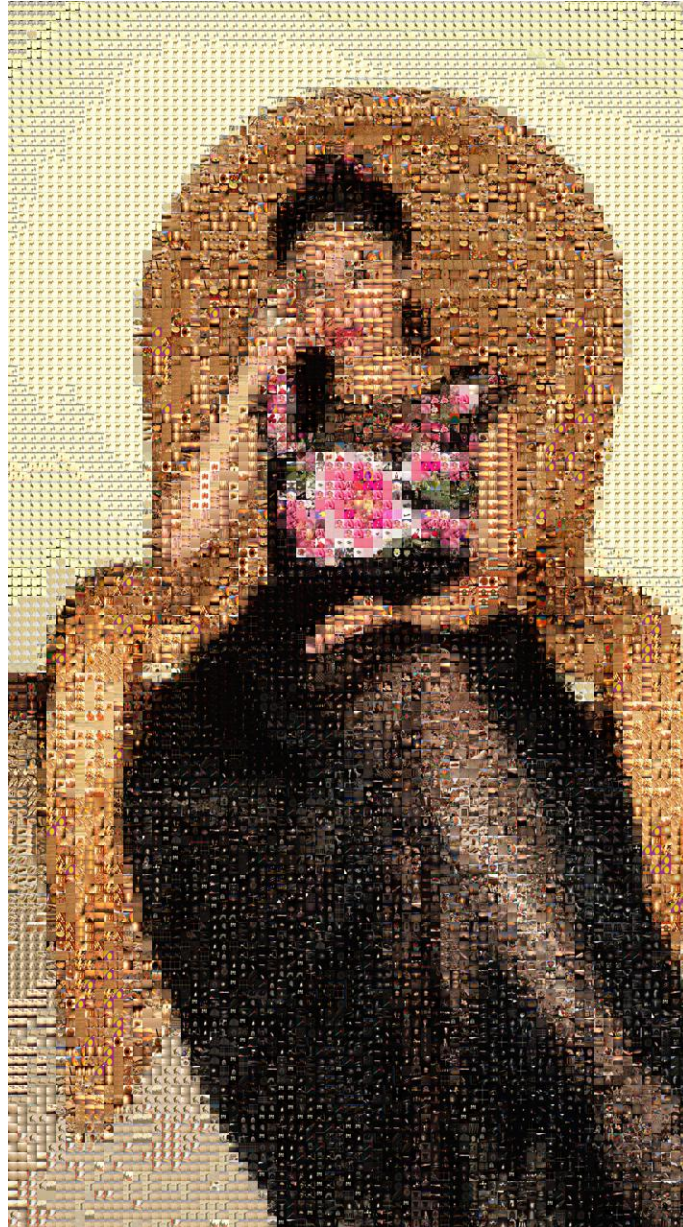


Figura 4: Imagen del Optenida.

Conclusión

Este programa es una herramienta versátil para crear fotomosaicos con un alto nivel de personalización. Su implementación de múltiples métodos de selección de imágenes permite al usuario experimentar con diferentes técnicas para optimizar el resultado visual.

Notas del Proyecto

La implementación del programa está pensada para reducir y aprovechar al máximo la memoria del procesador ya que me vi con la necesidad de ocupar menos recursos del CPU, el proceso de cada uno de los dos filtros que se implementaron en

aproximadamente 20 minutos en una laptop con las siguientes características:

```
1      edgar@edgar-Latitude-3380
2      -----
3      OS: Ubuntu 24.04.1 LTS x86_64
4      Host: SVF14A15CLB C10GT0SA
5      Kernel: 6.8.0-48-generic
6      Uptime: 2 hours, 31 mins
7      Packages: 4175 (dpkg), 47 (snap)
8      Shell: bash 5.2.21
9      Resolution: 1360x768
10     DE: GNOME 46.0
11     WM: Mutter
12     WM Theme: Adwaita
13     Theme: Yaru-magenta-dark [GTK2/3]
14     Icons: Yaru-magenta [GTK2/3]
15     Terminal: gnome-terminal
16     CPU: Intel i5-3337U (4) @ 2.700GHz
17     GPU: Intel 3rd Gen Core processor Graphics
        Controller
18     GPU: NVIDIA GeForce GT 735M
19     Memory: 4753MiB / 11839MiB
```