



Instituto Tecnológico de Costa Rica

Sede San Carlos

Unidad de Computación

Arquitectura de computadores

Primer proyecto : Collage imágenes

Profesor:

Luis Diego Gómez Rodríguez

Estudiantes:

María José Solís García

Evelyn Karina Urbina Álvarez

II Semestre 2022

Introducción

El siguiente proyecto tiene como objetivo la implementación de un algoritmo que permita crear a partir de diferentes imágenes en una lista, una imagen final formando un collage de una fotografía base, esto con el fin de conocer más acerca del procesamiento de imágenes basado en sus valores RGB por píxeles. Además, se aplica también microprocesamiento y paralelismo al producto final, lo anterior para comprender los conceptos que implican los mismos, así como reducir el tiempo de ejecución y recursos utilizados durante la creación del collage.

Análisis y descripción del problema

Según el problema planteado, se requieren resolver los siguientes puntos:

1)Procesamiento de imagen base y fotografías para el collage

- Para crear un collage de una imagen base a partir de muchas imágenes diferentes, se necesita, como anteriormente se mencionó, una foto base, y crear una lista de fotos. La imagen base, tendrá que ser una figura de alta calidad y se obtendrá llamando a su ruta de acceso, en cuanto a las demás imágenes, se necesita una carpeta que las contenga. Se necesita también que se agregue cada imagen dentro de la carpeta, y se almacene en una nueva lista para su procesamiento en las demás etapas de la implementación del algoritmo final.

2)Modificación del tamaño de las imágenes que formaran el producto final

- Se necesita recorrer la lista mencionada en el punto anterior, y obtener cada imagen de esta para que sea redimensionada a una dimensión mucho más diminuta que la original, esto con el fin de que de ese tamaño sea cada aparición de estas al formar el collage.

3) Cálculo de color predominante en cada imagen

- En este punto, se define el proceso base para el calculo de el color predominante en cada imagen dentro de las fotografías que se utilizaran para la construcción del retrato final. Para esto se necesita analizar cada píxel de la imagen, determinar el valor RGB predominante, y almacenar esta información en una lista diferente.

4) Comparar los pixeles de la foto con el color más parecido

- Toma la imagen y por cada píxel en ella, esto determinado por el tamaño de esta, retorna un valor, el cual sería el correspondiente al valor RGB del píxel seleccionado.

5)Formar la imagen final

- Según los valores obtenidos en el punto anterior, reemplaza cada píxel de la imagen base con la fotografía de la lista de imágenes

redimensionadas que contenga un color promedio al valor del píxel seleccionado. Guarda la imagen de reemplazo, en la posición del píxel, en una nueva imagen, y repite el proceso hasta completar la imagen base e imprimirla.

Solución del problema

Etapas 1:

En la primera etapa se le dará solución al procesamiento de la imagen base y fotografías para el collage.

Se define una variable la cual almacena la ruta de la imagen que será tomada como base para realizar el mosaico. Esta imagen puede ser cualquiera que el usuario desee, solo deberá anotar la ruta de esta y tendrá que ser en formato jpg. Se recomienda utilizar imágenes de alta calidad para un mejor resultado.

En cuanto a la carpeta de imágenes, para este proyecto, se almacena en una variable la ruta de esta, luego se crea una lista denominada "imagenesCollage", mediante un ciclo "for" y con ayuda de la librería "glob" se recorre la carpeta seleccionada, y cada objeto en esta se almacena dentro de la lista recientemente creada, finalizando lo propuesto para esta primera etapa.

Etapas 2:

La modificación del tamaño de las imágenes que formaran el producto final será el problema solucionado en la etapa dos. En este caso, se define una variable "tamañoMosaico" la cual almacena dos datos tipo int, luego se define una nueva lista denominada "imagenesListas", la cual almacenará las fotografías ya redimensionadas. Gracias a un ciclo for, se recorre la lista de imágenes para el collage, se abre cada imagen y mediante el método "resize" se modifica su tamaño enviándole como parámetro la variable "tamañoMosaico". La imagen modificada se almacena dentro de "imagenesListas", concluyendo esta etapa.

Etapas 3:

En cuanto al cálculo de color predominante en cada imagen, se utilizó el método, se crea una nueva lista denominada "coloresPíxeles" la cual almacenará los valores del color promedio de cada imagen procesada y obtenida mediante el recorrido de "imagenesListas" en un ciclo "for". Los valores que representan los colores son obtenidos gracias a la ayuda de la librería "numpy" y los métodos "array" y "mean". Luego de esto, los valores de la imagen se almacenan en la lista de "coloresPíxeles".

Etapas 4:

La cuarta etapa abarcará comparar los píxeles de la foto con el color más parecido, antes de empezar, se pixela la imagen base, dividiendo su tamaño entre el tamaño definido anteriormente para cada mosaico. Esto al igual que la redimensión de las imágenes anteriores, se realiza mediante el método “resize” enviándole como parámetros los enteros correspondientes al resultado de la división de su ancho entre el ancho del mosaico, y su altura, entre la altura de estos mismos.

Luego, para encontrar la imagen con el color más cercano al de un píxel, se utiliza un KDTree, para así encontrar los valores x,y,z que son equivalentes a los valores R,G y B del píxel, este KDTree se crea a partir de la lista de “coloresPíxeles”. Se crea una variable denominada “imagenColor” la cual corresponde a una matriz para almacenar índices.

Con un ciclo “for”, cuando “i” y “j” (correspondiente a la fila y columna del píxel), se encuentren dentro del rango del ancho y alto de la imagen base, tomará este píxel y analizará sus valores, para luego continuar con el reemplazo de píxeles por imágenes y la impresión del producto final. Y así sucede con cada píxel dentro de la imagen.

Etapas 5:

Finalmente, en esta etapa, se forma la imagen final, esto ocurre primero definiendo una nueva variable, denominada “imagenFinal”, la cual crea un nuevo archivo tipo jpg, a color y con el tamaño de la imagen base, luego, un ciclo “for”, cuando “i” y “j” (correspondiente a la fila y columna del píxel), se encuentren dentro del rango del ancho y alto de la nueva fotografía, según una posición “x”, “y” definida por la anchura del mosaico multiplicado por i, y la altura multiplicado por el valor de j, se imprime en ella la imagen de la lista colores, que más se parezca a los valores del píxel y se encuentre en la lista de imágenes listas.

Cuando ya no haya más píxeles de la base que reemplazar, se guarda la imagen final y por último, se abre y se muestra la imagen en pantalla al usuario.

Resultados:

Las siguientes tablas muestran resultados obtenidos de las diferentes pruebas de tiempo de ejecución, para diferentes imágenes base, específicamente las **estadísticas de procesamiento secuencial**

Tabla 1: Imagen de la chica

Dimensión de la imagen: 1920x1080	
Proceso	Tiempo de ejecución
Buscar las rutas de las imágenes	0.000000 segundos
Obtener las imágenes de la colección de imágenes	0.005016 segundos
Modificar el tamaño de las imágenes	4.967054 segundos
Obtener los colores promedios de las imágenes	0.041701 segundos
Cambiar el tamaño de la imagen base del collage	0.049702 segundos
Encontrar las imágenes que coincide con cada de píxel la imagen base	0.549080 segundos
Ensamblar la nueva imagen	0.090575 segundos
Mostrar la nueva imagen	2.181109 segundos
Creación total del collage	7.886252 segundos

Fuente: Creación propia

Tabla 2: Imagen de un gato

Dimensión de la imagen: 2303x3012	
Proceso	Tiempo de ejecución
Buscar las rutas de las imágenes	0.000000 segundos
Obtener las imágenes de la colección de imágenes	0.005072 segundos
Modificar el tamaño de las imágenes	4.399089 segundos
Obtener los colores promedios de las imágenes	0.044664 segundos
Cambiar el tamaño de la imagen base del collage	0.133406 segundos
Encontrar las imágenes que coincide con cada de píxel la imagen base	0.329171 segundos
Ensamblar la nueva imagen	0.121407 segundos
Mostrar la nueva imagen	2.710996 segundos
Creación total del collage	7.748417 segundos

Fuente: Creación propia

Tabla 3: Imagen de un girasol

Dimensión de la imagen: 3648x4934	
Proceso	Tiempo de ejecución
Buscar las rutas de las imágenes	0.000000 segundos
Obtener las imágenes de la colección de imágenes	0.005066 segundos
Modificar el tamaño de las imágenes	4.461949 segundos
Obtener los colores promedios de las imágenes	0.040724 segundos
Cambiar el tamaño de la imagen base del collage	0.260997 segundos

Encontrar las imágenes que coincide con cada de píxel la imagen base	0.851832 segundos
Ensamblar la nueva imagen	0.284783 segundos
Mostrar la nueva imagen	3.208055 segundos
Creación total del collage	9.116393 segundos

Fuente: Creación propia

Tabla 4: Imagen de Luffy

Dimensión de la imagen: 7680x4320	
Proceso	Tiempo de ejecución
Buscar las rutas de las imágenes	0.000000 segundos
Obtener las imágenes de la colección de imágenes	0.005836 segundos
Modificar el tamaño de las imágenes	5.452352 segundos
Obtener los colores promedios de las imágenes	0.049868 segundos
Cambiar el tamaño de la imagen base del collage	0.413479 segundos
Encontrar las imágenes que coincide con cada de píxel la imagen base	1.345340 segundos
Ensamblar la nueva imagen	0.501305 segundos
Mostrar la nueva imagen	5.604025 segundos
Creación total del collage	13.376719 segundos

Fuente: Creación propia

Tabla 5: Imagen de One Piece

Dimensión de la imagen: 2303x3012	
Proceso	Tiempo de ejecución
Buscar las rutas de las imágenes	0.000000 segundos
Obtener las imágenes de la colección de imágenes	0.004762 segundos
Modificar el tamaño de las imágenes	4.891468 segundos
Obtener los colores promedios de las imágenes	0.040191 segundos
Cambiar el tamaño de la imagen base del collage	0.463415 segundos
Encontrar las imágenes que coincide con cada de píxel la imagen base	1.236367 segundos
Ensamblar la nueva imagen	0.493497 segundos
Mostrar la nueva imagen	5.834156 segundos
Creación total del collage	12.966788 segundos

Fuente: Creación propia

Análisis de resultados

A continuación, se muestra una tabla que incluye el nombre, estado y observaciones de los resultados obtenidos durante la creación e implementación en Python de las etapas descritas en la solución del problema.

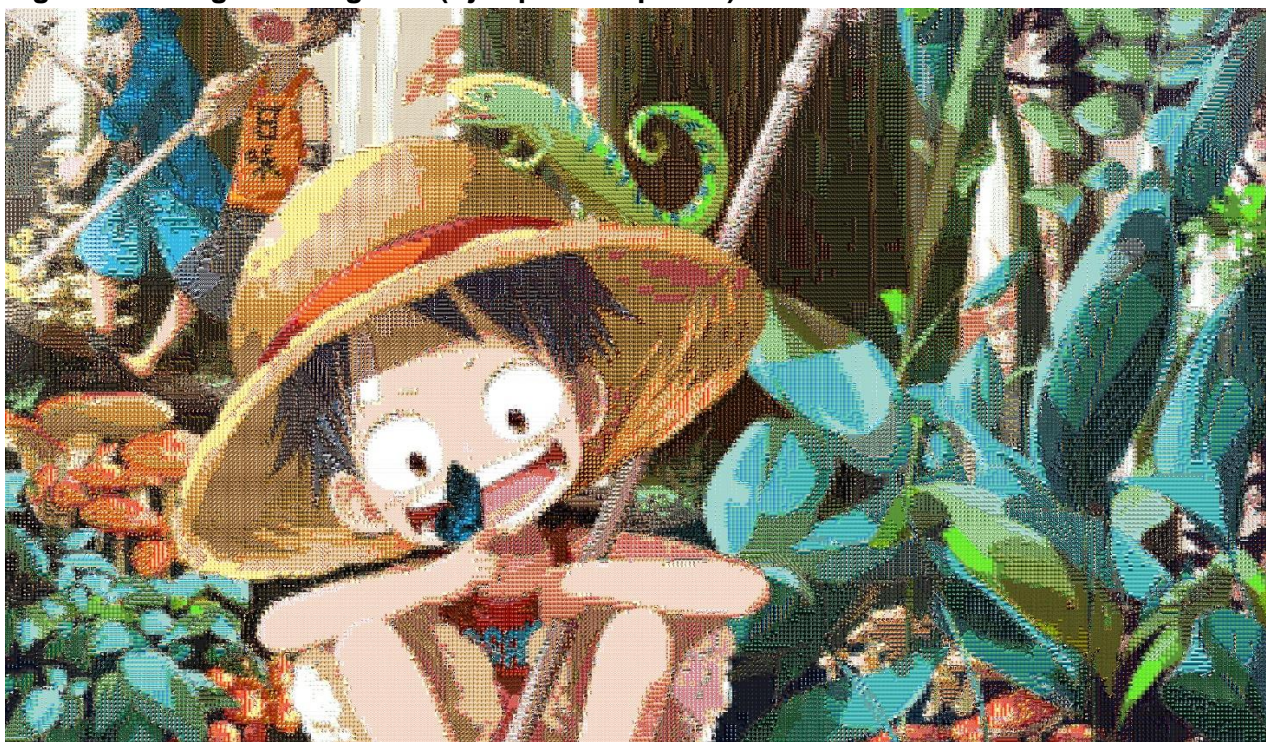
Tarea	Estado	Observaciones
Procesamiento de imagen base y fotografías para el collage	Completo	En el código implementado, se da el funcionamiento correcto de todas las propuestas planteadas en la solución del problema.
Modificación del tamaño de las imágenes que formaran el producto final	Completo	El tamaño de las imágenes cambia correctamente.
Cálculo de color predominante en cada imagen	Completo	Su ejecución es correcta y todos son bien calculados y almacenados.
Comparar los pixeles de la foto con el color más parecido	Completo	Su implementación es correcta como se describió en la solución del problema.
Formar la imagen final	Completo	Todo funciona tal y como se esperaba según lo planteado (ver Figura1 , Figura2 , Figura3)

Figura 1: Imagen Original(Ejemplo “onepiece”)



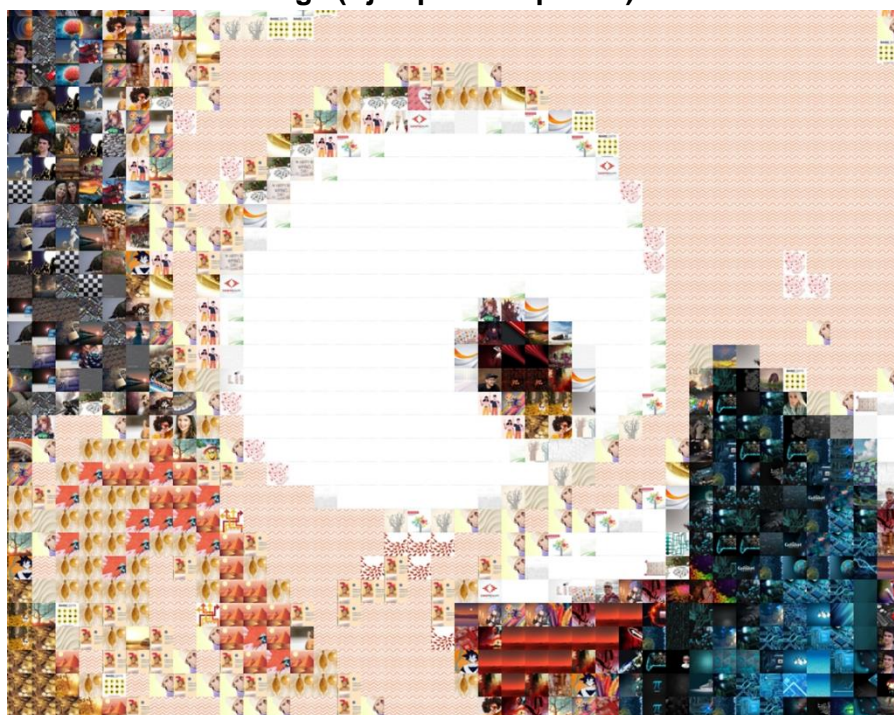
Fuente: Wallpaperbetter.com

Figura 2: Collage de imágenes (Ejemplo “onepiece”)



Fuente: Creación Propia

Figura 3: Acercamiento del Collage (Ejemplo “onepiece”)



Conclusiones

Luego de la implementación del código y el análisis de los resultados, se concluye que:

- El programa cumple las expectativas en cuanto al procesamiento de imágenes, obtención de valores RGB, y reemplazo de pixeles de la imagen base por las fotografías seleccionadas.
- El programa en un futuro puede llegar a estar funcionando mejor si se aplican diferentes métodos de ahorro de recursos y mejor paralelismo en la ejecución del software.

Referencias

Gómez, L. (19 de Octubre de 2022). *Intro Colab IIS 2022*. Obtenido de <https://colab.research.google.com/drive/14ECSwDYN6x9R4GqXve37JG2irjeTn4xP>

Rvziel. (22 de Mayo de 2017). *Extraccion de datos (valores de pixel) de una imagen con Python*. Obtenido de <https://pythoneyes.wordpress.com/2017/05/22/extraccion-de-datos-valores-de-pixel-de-una-imagen/>

The SciPy community. (19 de October de 2022). *SciPy documentation*. Obtenido de <https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.cKDTree.html>

