

Manipulación de imágenes

Baños Islas Jesús Alberto
9CV12

INTRODUCCIÓN

En el ámbito del procesamiento de imágenes, Python se destaca como una herramienta versátil y eficiente, especialmente con el uso de bibliotecas como NumPy y OpenCV (cv2). Estas bibliotecas facilitan una amplia gama de operaciones con imágenes, desde manipulaciones básicas hasta complejas transformaciones.

NumPy ofrece soporte para grandes arreglos y matrices multidimensionales, junto con una colección de funciones matemáticas para trabajar con estas estructuras. En el procesamiento de imágenes, NumPy se utiliza principalmente para la manipulación y operación de píxeles.

OpenCV (cv2): Una Herramienta Potente para el Procesamiento de Imágenes.

OpenCV es una biblioteca de código abierto destinada a aplicaciones de visión por computadora y procesamiento de imágenes. Proporciona herramientas para leer y escribir imágenes, manipular colores y realizar diversas transformaciones.

Suma de Imágenes: La suma implica aumentar la intensidad de los píxeles combinando dos imágenes. Diferente a la suma regular, OpenCV maneja la saturación de manera eficiente.

Resta de Imágenes: Esta operación ayuda a resaltar las diferencias entre dos imágenes, restando los valores de píxeles de una imagen de otra.

Multipliación de Imágenes: Al multiplicar imágenes, se ajustan las intensidades de los píxeles, lo que puede ser útil para ajustar la exposición o aplicar máscaras.

Inversión de Imágenes: Útil para correcciones de orientación o para efectos visuales.

Transpuesta de Imágenes: Al transponer, rotamos la imagen en un ángulo de 90 grados, lo cual es útil en correcciones de orientación o para efectos visuales.

Separación de Canales RGB: Esta operación es crucial para trabajar con colores específicos en una imagen, lo que permite manipulaciones detalladas a nivel de canal.

IMÁGENES COMO ARREGLOS TRIDIMENSIONALES

Las imágenes a color en el ámbito digital se pueden conceptualizar como matrices tridimensionales, una forma de representación que es clave en el procesamiento de imágenes. Este concepto se fundamenta en el modelo RGB (Red, Green, Blue), que es el estándar para definir colores en términos de la intensidad de rojo, verde y azul.

En esta representación, la imagen es vista como una matriz tridimensional. Las primeras dos dimensiones de esta matriz corresponden a las dimensiones espaciales de la imagen: su altura y anchura, representadas en el número de píxeles. La tercera dimensión se refiere a los componentes del color. Cada píxel en la imagen, por lo tanto, se compone de un trío de valores que determinan la intensidad de los colores rojo, verde y azul.

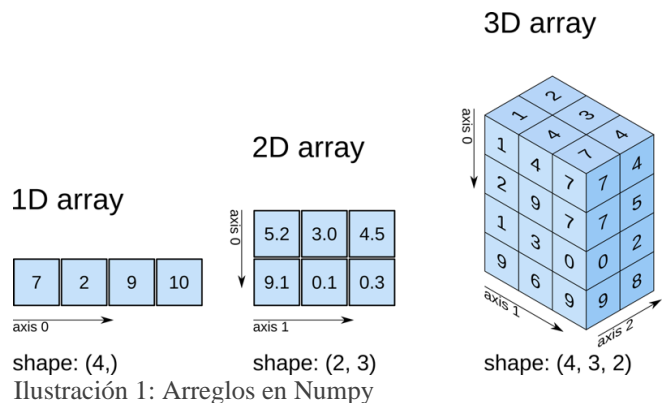


Ilustración 1: Arreglos en NumPy

La manipulación de estas matrices tridimensionales permite realizar una amplia variedad de procesamiento en las imágenes. Esto incluye desde ajustes básicos de color hasta operaciones más complejas como filtros y transformaciones geométricas. Por ejemplo, alterar los valores asociados al rojo en toda la matriz puede cambiar la tonalidad rojiza de la imagen entera.

Esta manera de representar las imágenes tiene una importancia crucial en numerosos campos, incluyendo la visión por computadora, el diseño gráfico, la edición de imágenes, y en áreas de inteligencia artificial que se enfocan en el reconocimiento y procesamiento de imágenes.

Desarrollo

I. IMAGEN CUADRICULADA

Se implementó un algoritmo para manipular imágenes digitales utilizando una librería de procesamiento de imágenes. El algoritmo comenzó con la lectura de una imagen. Una vez cargada la imagen, se extrajeron sus dimensiones, incluyendo el alto, el ancho y la profundidad de los canales de color, y se almacenaron en variables separadas.

A continuación, se llevó a cabo un redimensionamiento de la imagen a un tamaño específico, en este caso, 1280x720 píxeles. Este paso es crucial para estandarizar el tamaño de la imagen para las operaciones subsiguientes.

Se realizó un bucle que recorrió la imagen horizontalmente a intervalos regulares (cada 50 píxeles). En cada paso de este bucle, se dibujó una línea horizontal en la imagen. De manera similar, dentro de este bucle, se ejecutó otro bucle que recorrió la imagen verticalmente, también a intervalos de 50 píxeles. En cada paso de este bucle anidado, se dibujó una línea vertical. Las líneas se trazaron en un color contrastante, típicamente negro, y con un grosor de un píxel. Este proceso creó un patrón de cuadrícula sobre la imagen.

DIAGRAMA DE FLUJO

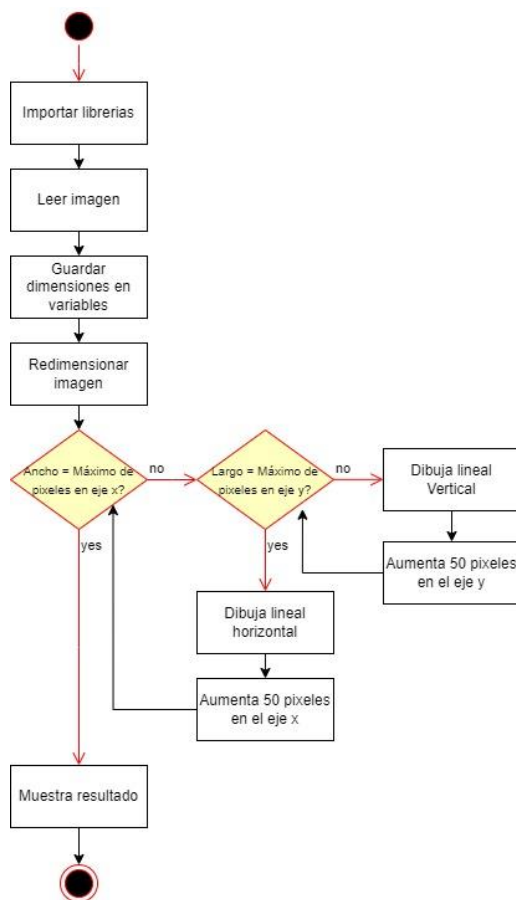


Imagen utilizada para la prueba



Ilustración 2: fotografía tomada en el salón

RESULTADO

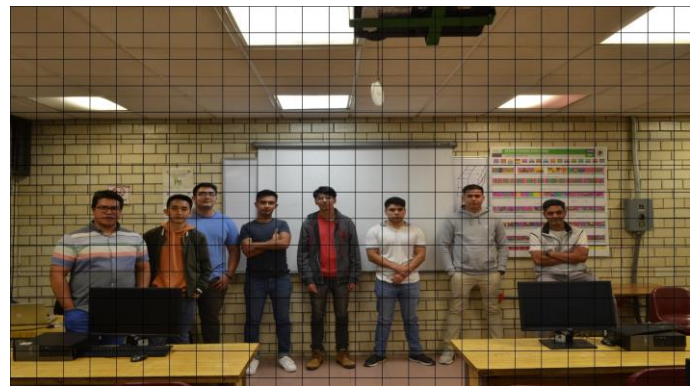


Ilustración 3: Resultado cuadrículado

II. SUMA DE IMÁGENES

Se realiza la suma de dos imágenes digitales. Este proceso comienza con la carga de dos imágenes diferentes desde archivos específicos. Una vez cargadas estas imágenes, se ajustan ambas al mismo tamaño. Este paso es crucial para garantizar que las dimensiones de ambas imágenes sean idénticas, permitiendo así una operación de suma pixel a pixel.

Tras el redimensionamiento, se calcula las dimensiones de las imágenes ajustadas, que incluyen el alto, el ancho y la profundidad de los canales de color. Se crea una nueva imagen, inicialmente vacía (o negra), con las mismas dimensiones.

El algoritmo define dos factores de ponderación, A y B, que se utilizan para determinar la contribución relativa de cada imagen en la suma final. Estos factores son esenciales para controlar la visibilidad de cada imagen en la imagen resultante.

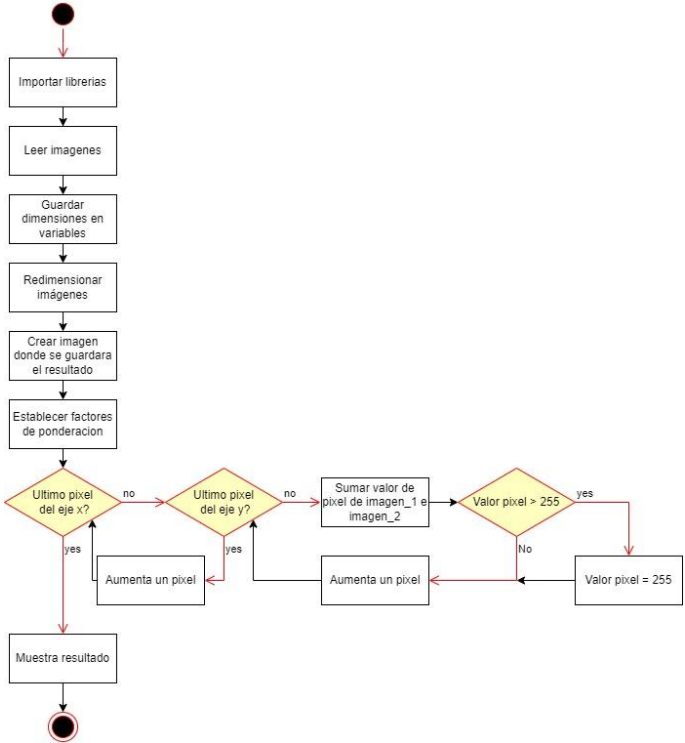
Se inicia un proceso iterativo que recorre cada pixel de las imágenes. En cada iteración, se suman los valores de los píxeles correspondientes de las dos imágenes, aplicando los factores de ponderación A y B. Si la suma ponderada de los valores de los píxeles es menor o igual a 255 (el valor máximo para un canal de color en una imagen de 8 bits), este valor se asigna directamente al pixel correspondiente en la imagen resultante. Si la suma excede 255, se asigna el valor máximo de 255 a dicho pixel para evitar la saturación del color.

Una vez completado el proceso de suma para todos los píxeles, se muestra la imagen resultante en una ventana de visualización. Esta ventana permanece activa hasta que se detecta una acción de entrada del usuario, como la presión de una tecla. Tras esta acción, se cierra la ventana y finaliza el procesamiento.

Este enfoque demuestra un método para combinar dos imágenes en una sola, permitiendo la superposición y mezcla de contenido visual de ambas imágenes de manera controlada.

Se realizaron múltiples pruebas haciendo variar los factores de ponderación A y B para poder ver como es que estos valores afectan directamente al resultado final de la suma.

DIAGRAMA DE FLUJO



Los valores considerados fueron los siguientes:

A	B
0.2	0.8
0.4	0.6
0.5	0.5
0.6	0.4
0.8	0.2

Las imágenes utilizadas fueron las siguientes:



Ilustración 4 Primer fotografía tomada en el salón



Ilustración 5: Segunda fotografía tomada en el salón

RESULTADOS

Para $A = 0.2$ y $B = 0.8$



Ilustración 6: Primer resultado de suma de imágenes con valores de A y B previamente definidos

Para $A = 0.4$ y $B = 0.6$



Ilustración 7: Segundo resultado de suma de imágenes con valores de A y B previamente definidos

Para $A = 0.5$ y $B = 0.5$



Ilustración 8: Tercer resultado de suma de imágenes con valores de A y B previamente definidos

Para $A = 0.6$ y $B = 0.4$



Ilustración 9: Cuarto resultado de suma de imágenes con valores de A y B previamente definidos

Para $A = 0.8$ y $B = 0.2$



Ilustración 10: Quinto resultado de suma de imágenes con valores de A y B previamente definidos

III. RESTA DE IMÁGENES

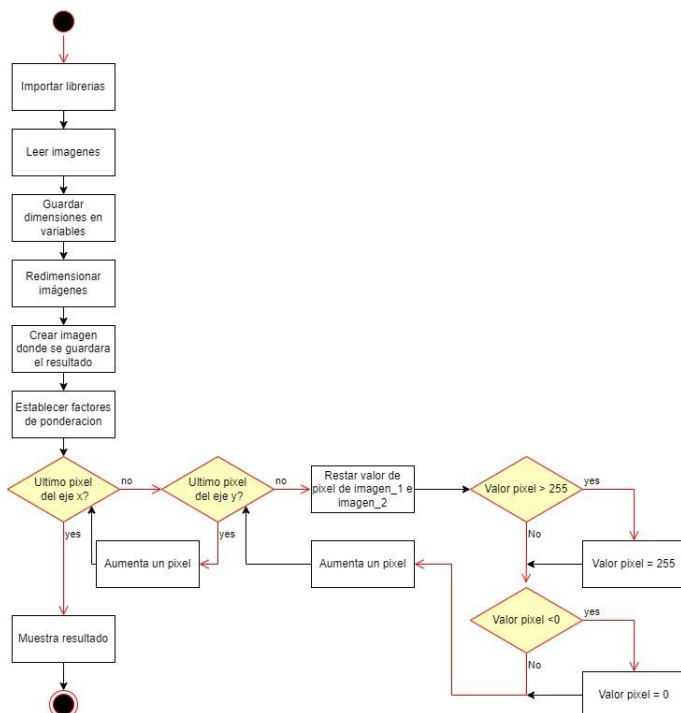
El programa comienza leyendo dos imágenes desde archivos específicos. Estas imágenes son convertidas en matrices, que permiten realizar operaciones matemáticas con sus píxeles.

Una vez cargadas, el programa determina las dimensiones de las imágenes, incluyendo la altura, la anchura y la profundidad de los canales de color. Con estas dimensiones, se crea una nueva imagen en blanco (matriz de ceros) del mismo tamaño que las imágenes originales.

Posteriormente entra en un proceso iterativo, donde recorre cada píxel de las imágenes. En cada píxel, se realiza una operación de resta entre los valores de color correspondientes de las dos imágenes originales. Si el resultado de esta resta es negativo, se ajusta a cero, y si es mayor que 255, se limita a 255. Estos límites aseguran que los valores de los píxeles permanezcan dentro del rango válido para una imagen en formato de color estándar.

El resultado de estas operaciones de resta se almacena en la nueva imagen, creando así una imagen que es el resultado de la diferencia entre las dos imágenes originales. Finalmente, el programa muestra la imagen resultante en una ventana y espera hasta que se presione una tecla. Al recibir la entrada del usuario, el programa cierra todas las ventanas abiertas y finaliza su ejecución.

DIAGRAMA DE FLUJO



Las imágenes utilizadas fueron las siguientes:



Ilustración 11: Primera imagen a restar



Ilustración 12: Segunda imagen a restar

RESULTADOS



Ilustración 13: Resultado de la resta

Al restar los valores de píxeles de dos imágenes y ajustar el resultado para que se mantenga dentro del rango estándar de 0 a 255, el programa efectivamente resalta las diferencias entre ambas imágenes. Este método de sustracción de píxeles y posterior limitación asegura que cualquier variación en color o intensidad entre las dos imágenes originales se magnifique en la imagen resultante. Así, las áreas donde las imágenes difieren se hacen más evidentes, mientras que las regiones donde son similares tienden a neutralizarse.



Ilustración 14: Se resaltan los resultados de la resta

IV. MULTIPLICACIÓN DE IMÁGENES

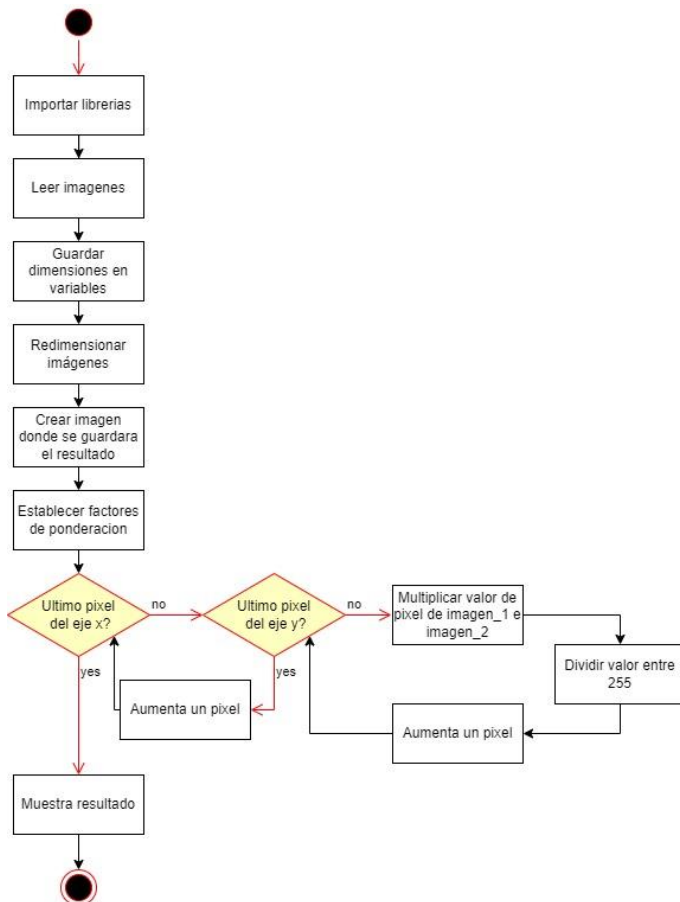
Se inicia el proceso leyendo dos imágenes diferentes desde archivos. Cada imagen es redimensionada a un tamaño uniforme de 1280x720 píxeles, estandarizando así las dimensiones para las operaciones subsiguientes.

Una vez redimensionadas, las imágenes son convertidas en matrices, facilitando así la realización de operaciones matemáticas sobre los píxeles. Estas matrices se utilizan para almacenar y manipular los datos de color de las imágenes.

Posteriormente se procede a crear una nueva imagen, inicialmente en negro, del mismo tamaño que las imágenes redimensionadas. A continuación, se ejecuta un proceso iterativo sobre cada píxel de las imágenes. En cada iteración, los valores correspondientes de los píxeles de ambas imágenes se multiplican entre sí, y el resultado se divide por 255. Este paso es crucial para mantener los valores resultantes dentro del rango de color estándar.

El resultado de esta operación de multiplicación es almacenado en la nueva imagen, generando así una imagen que es el producto de la combinación de las dos imágenes originales. Este proceso efectivamente mezcla los datos de color de ambas imágenes de una manera única.

DIAGRAMA DE FLUJO



Las imágenes utilizadas fueron las siguientes:



Ilustración 15: Primer fotografía tomada en el salón



Ilustración 16: Segunda fotografía tomada en el salón

RESULTADOS



Ilustración 17: Resultado de realizar la multiplicación

La multiplicación de imágenes, sirve para fusionar visualmente dos imágenes de una manera que resalta las características comunes y modifica las diferencias. Este método produce una imagen resultante donde los elementos similares en ambas imágenes se intensifican, mientras que las diferencias se atenúan o transforman.

V. IMAGEN TRANSPUESTA

Se comienza cargando una imagen específica desde un archivo. Luego, ajusta el tamaño de esta imagen a un formato cuadrado de 500x500 píxeles, estandarizando así sus dimensiones para el procesamiento posterior.

Una vez redimensionada, la imagen es mostrada en una ventana, permitiendo al usuario visualizarla antes de proceder con la siguiente operación. La clave del programa es la creación de una nueva imagen, inicialmente vacía, del mismo tamaño que la imagen original.

El siguiente paso es un proceso iterativo en el que se recorren todos los píxeles de la imagen redimensionada. Durante esta iteración, se realiza una operación de transposición de la imagen: cada píxel de la imagen original se coloca en una nueva posición en la imagen vacía, intercambiando sus coordenadas 'x' e 'y'. Esto significa que lo que estaba en la fila se mueve a la columna correspondiente y viceversa.

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \quad A^T = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$$

Ilustración 18: Transpuesta de una matriz

Una vez completada la transposición, la nueva imagen resultante se muestra en otra ventana. Esta imagen transpuesta presenta una orientación alterada en comparación con la original, efectivamente reflejando la imagen a lo largo de su diagonal.

DIAGRAMA DE FLUJO

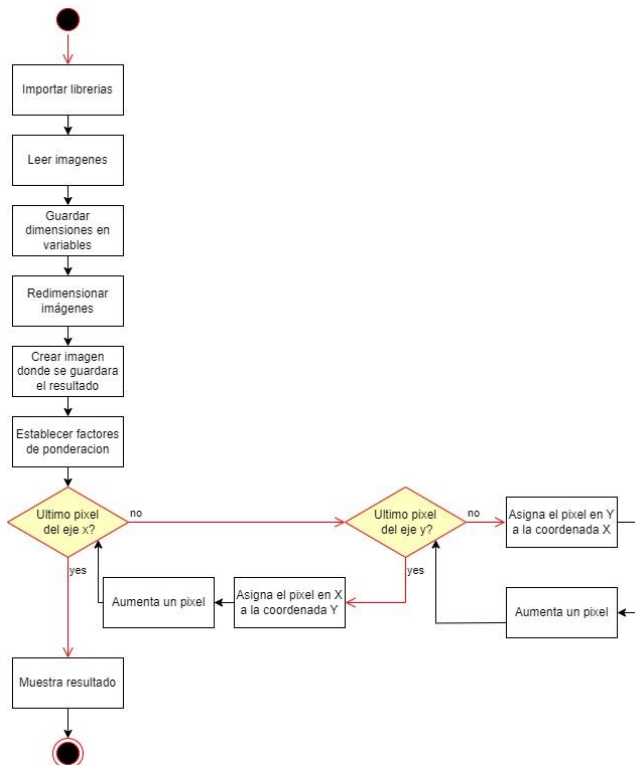


Imagen utilizada para la prueba



Ilustración 19: fotografía tomada en el salón

RESULTADOS

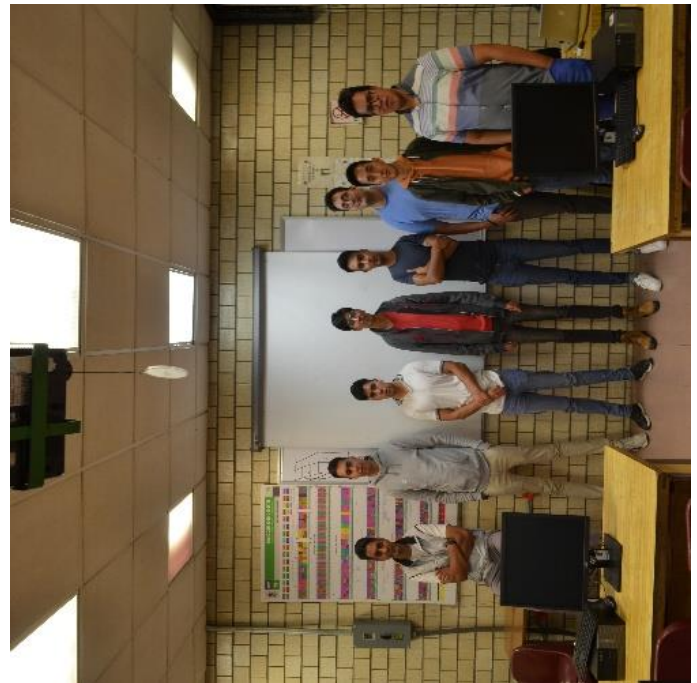


Ilustración 20: Resultados tras aplicar la transpuesta

VI. IMAGEN INVERSA

Se inicia su proceso cargando una imagen desde un archivo, que luego es redimensionada a un tamaño específico de 500x500 píxeles. Esta acción uniformiza las dimensiones de la imagen, preparándola para el procesamiento subsiguiente.

Una vez que la imagen ha sido redimensionada, se muestra al usuario en una ventana, permitiéndole observar la imagen original antes de la manipulación. Luego, el programa procede a calcular las dimensiones de la imagen redimensionada, que son esenciales para el proceso de inversión.

Se crea una nueva imagen, inicialmente en blanco, con las mismas dimensiones que la imagen original. El programa entra en un bucle iterativo que recorre la imagen redimensionada de manera inversa. En cada paso de este bucle, los píxeles de la imagen original se asignan a posiciones inversas en la nueva imagen. Esencialmente, los píxeles que están en la parte superior derecha de la imagen original se mueven a la parte inferior izquierda de la nueva imagen, y viceversa.

Este proceso de inversión altera la orientación de la imagen, creando un efecto de espejo tanto horizontal como vertical. Una vez completada la inversión, la nueva imagen resultante se muestra en otra ventana, permitiendo al usuario compararla con la original.

DIAGRAMA DE FLUJO

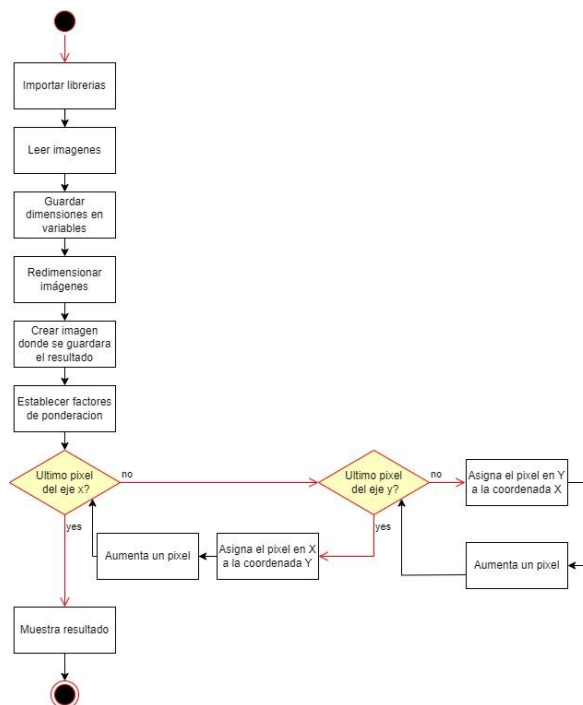


Imagen utilizada para la prueba



Ilustración 21: fotografía tomada en el salón

RESULTADOS

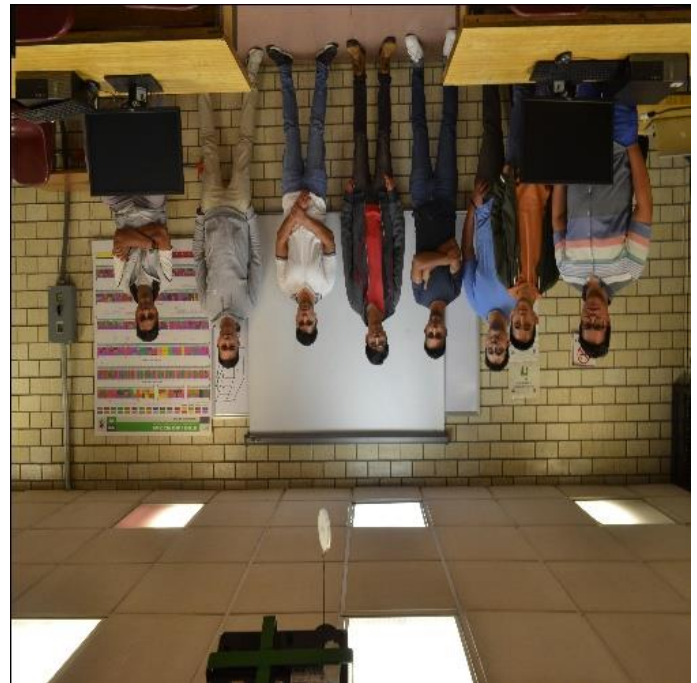


Ilustración 22: Resultado de invertir los pixels

La creación de una imagen inversa es una técnica valiosa en el procesamiento de imágenes digitales. Al invertir la orientación de la imagen tanto horizontal como verticalmente, se genera un efecto espejo que puede ser útil tanto para análisis visuales como para aplicaciones de diseño gráfico.

CONCLUSIÓN GENERAL

El procesamiento de imágenes es una disciplina de gran importancia en el mundo tecnológico moderno, que se beneficia enormemente de herramientas y bibliotecas especializadas. Estas herramientas proporcionan capacidades esenciales para manipular y transformar imágenes digitales de manera efectiva y eficiente.

La importancia de este campo radica en su versatilidad y aplicación en numerosos sectores. Por ejemplo, en el ámbito médico, el procesamiento de imágenes es crucial para mejorar y analizar imágenes médicas, como rayos X o resonancias magnéticas, lo que ayuda en diagnósticos precisos y planificación de tratamientos. En la industria del entretenimiento, se utiliza para la creación de efectos visuales y animaciones en películas y videojuegos. Además, en el sector de la seguridad, el procesamiento de imágenes es fundamental para el análisis y reconocimiento facial en sistemas de videovigilancia.

Una herramienta clave en el procesamiento de imágenes es una biblioteca especializada en el manejo de grandes arreglos y matrices multidimensionales. Esta biblioteca facilita operaciones matemáticas complejas con estructuras de datos, permitiendo manipular imágenes a nivel de píxel. Esto es esencial para ajustar imágenes, mejorar su calidad, o cambiar su orientación y tamaño.

REFERENCIAS

- [1] Nilss J., Nilsson, Principios de Vision por Computadora, Díaz de Santos, España, 1987, 110-270pp.
- [2] Rich E, Vision por Computadora, McGraw-Hill, España, 1991, 100-340pp.
- [3] Parker J. R. , Algorithms for Image Processing and Computer Vision, John Wiley & Sons, USA, 1997, 432pp.