

Indicaciones generales

- Recuerda que la **tarea es individual**. Los casos de copia/plagio serán sancionados con nota cero (0) en la asignatura.
- La tarea será publicada en Canvas pero **se entrega por la plataforma Gradescope**, además debe considerar:
 - Se le proporcionará un archivo solution.py. Usted deberá editar lo en las secciones correspondientes.
 - Al finalizar, solo subir el archivo solution.py (**NO cambiar el nombre del archivo y NO comprimirlo**).
 - Cada pregunta tiene casos de prueba, evalúe su solución con cada caso.

Gradescope

1. Nosotros les proporcionaremos un código base de donde deberán partir para completar dicho ejercicio. Este archivo es llamado **solution.py** y lo encontrarán en la indicación de la tarea en CANVAS.
2. Al finalizar, **solo** subir el archivo **solution.py** (**NO cambiar el nombre del archivo y NO comprimirlo**).
3. Cada pregunta tiene diversos casos de prueba. Para obtener la nota completa en una pregunta, el algoritmo debe obtener la respuesta correcta en dichos casos de prueba.
4. Si un caso de prueba falla, visualizarán un mensaje de error con sugerencias. **Lee el error**, revisa el código e inténtalo de nuevo.
5. Los input de los casos de prueba son confidenciales.

Indicaciones específicas

1. Ustedes deben escribir dentro de la sección y a la misma altura de donde está escrito "*SU SOLUCION EMPIEZA AQUI*". Además, no deben modificar nada debajo de "*SU SOLUCION TERMINA AQUI*". Recuerden tener cuidado con las indentaciones.
2. Los input del ejercicio se encuentran en la plantilla. Recuerden usar estas variables para resolver el ejercicio.

3. Ustedes podrán utilizar la imágenes en la carpeta llamada 'imagenes' para probar su implementación. Es importante que las imágenes las muevan a la carpeta donde se encuentra el archivo 'solution.py'. Si desean probar con una imagen externa, verificar que se encuentre en formato BMP de 24 bits.
4. La respuesta de los ejercicios debe ser retornada en una lista de tres dimensiones según se encuentra especificado en la plantilla otorgada.
5. Ustedes podrán utilizar las funciones de apoyo llamadas `leer_imagen` y `guardar_imagen` con la finalidad de convertir una imagen en formato BMP a una lista tridimensional y de una lista tridimensional a una imagen BMP.
6. Si realizan cálculos que conllevan decimales, utilizar la función `round` y convertirlo a entero antes de guardarla en la lista resultado.

Introducción a filtros de imágenes

Tu labor en esta ocasión es implementar un programa que aplique filtros a imágenes en formato BMP. Para ello crearemos 3 filtros.

Contexto

Una manera sencilla de representar imágenes es utilizando una cuadrícula de píxeles, los cuales pueden ser de distintos colores. Para una imagen en blanco y negro necesitamos solamente 1 bit, pues el 0 representa el negro y el 1 el blanco.

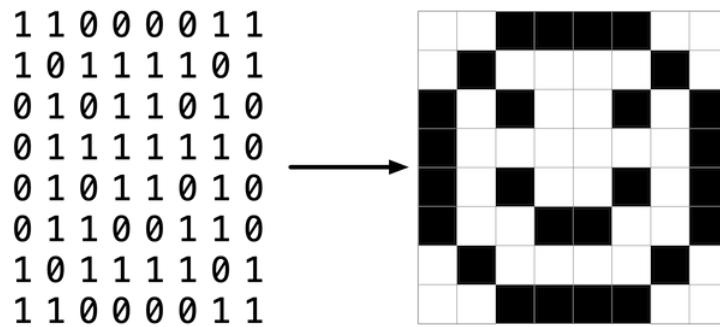


Figura 1: Representación de una imagen en blanco y negro

Ahora pensemos en imágenes más coloridas, para obtener más colores necesitaremos más bits por píxel que los representen. Formatos como BMP, JPEG y PNG soportan colores de *24 bits*. En realidad, BMP soporta colores de 1, 4, 8, 16, 24 y 32 bits. Pero en esta ocasión utilizaremos 24.

Con una imagen tipo BMP de 24 bits representaremos colores RGB: rojo, verde y azul (red, green and blue). Por lo que tenemos 8 bits que representan la cantidad de color de cada uno. 8 rojos + 8 verdes + 8 azules. Esto quiere decir que el valor de cada canal de calor (R,G,B) es como mucho 255.

Filtros

Ahora pensemos en los filtros, estos son modificaciones a las imágenes. Para realizarlos estaremos modificando sus píxeles de tal manera que genere el efecto deseado en la imagen.

Utilizaremos esta imagen como referencia:



Figura 2: Representación de la imagen original

Pregunta 1: Filtro alternativo - 4 pts

Para este ejercicio, alternaremos los colores de cada píxel de la imagen. Para ello, por cada canal (R,G,B) de cada píxel de la imagen realizaremos la operación $255 - \text{canal}$.

$$\text{pixel}[i][j][c] = 255 - \text{pixel}[i][j][\text{canal}] \quad 0 \leq \text{canal} < 3 \quad (1)$$

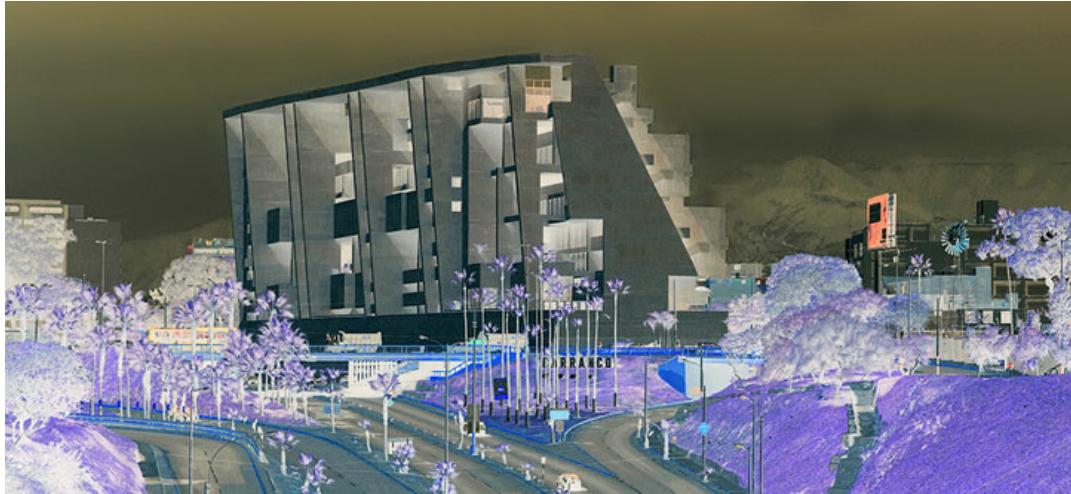


Figura 3: Representación de la imagen modificada con colores alternativos

Pregunta 2: Reflejar imagen - 4 pts

Para este problema, lo que deseamos hacer es realizar un reflejo de la imagen original, de manera que los píxeles de la izquierda se coloquen a la derecha y viceversa.



Figura 4: Representación de la imagen reflejada

Pregunta 3: Superposición - 6 pts

Para este ejercicio, recibiremos dos imágenes y una posición P . La primera imagen se colocará encima de la segunda imagen, donde la posición del píxel que se encuentra en la esquina superior izquierda de la primera imagen debe coincidir con P .



Figura 5: Representación de la imagen superpuesta en la posición (0,0)



Figura 6: Representación de la imagen superpuesta en la posición (200,500)

Pregunta 4: Similitud de imágenes - 6 pts

El objetivo del ejercicio es conocer el porcentaje de similitud entre dos imágenes. Para esto, tendremos de entrada dos imágenes y un factor de similitud. Se considera que dos píxeles son similares si su distancia euclídea es menor que el factor de similitud.

Sean las imágenes de dimensiones NxM imagen_1 e imagen_2, el píxel i,j de ambas imágenes se consideran similares si y solo si se cumple la siguiente condición:

$$\sqrt{\sum_{canal=0}^2 (imagen_1[i][j][canal] - imagen_2[i][j][canal])^2} < factor \quad 0 \leq i < N \wedge 0 \leq j < M \quad (2)$$

lo mismo que decir

$$dist(imagen_1[i][j], imagen_2[i][j]) < factor \quad \forall i, j \in \mathbb{Z} \quad 0 \leq i < N \wedge 0 \leq j < M \quad (3)$$

Considere que cada vez que se cumple esta condición la variable *pixeles_similares* aumenta en 1.

Finalmente, para conocer el porcentaje de similitud entre las dos imágenes usaremos la siguiente fórmula:

$$Porcentaje_de_similitud = round\left(\frac{pixeles_similares}{total_de_pixeles} * 100\right) \quad (4)$$

Note que la variable *total_de_pixeles* está dado por el producto de NxM.

NOTA:

- Utilizar la función round de Python.
- Las dimensiones de ambas imágenes son iguales
- Si deseas probar tu solución, utiliza las imágenes *foto_utec.bmp* y *foto_utec_r.bmp*

Ejemplo:

$$factor_de_similitud = 20$$



Figura 7: Representación de la imagen 1



Figura 8: Representación de la imagen 2

$$Porcentaje_de_similitud = 18 \quad (5)$$