

# Universidad de Guanajuato

DIVISIÓN DE CIENCIAS E INGENIERÍAS

CAMPUS LEÓN

## Fundamentos de Procesamiento Digital de Imágenes

Grupo:A

### Tarea 4 Operaciones Básicas Unidad 1

FECHA DE ENTREGA: 17 DE SEPTIEMBRE DE 2022

**Alumno:**

Miguel Ángel Hernández Tapia

**Profesor:**

Dr. Arturo González Vega

# Índice

<b>1. Introducción</b>	<b>3</b>
<b>2. Metodología</b>	<b>3</b>
<b>3. Resultados</b>	<b>4</b>
<b>4. Discusiones</b>	<b>6</b>
<b>5. Conclusión</b>	<b>7</b>
<b>A. Código</b>	<b>7</b>

# Fundamentos de Procesamiento Digital de Imágenes

## Tarea 4 Operaciones Básicas

Miguel Ángel Hernández Tapia

17 de septiembre de 2022

### Resumen

En este documento se presentan las operaciones básicas, negación, ajuste y transformación gamma. También se cambian las escalas a partir de un mínimo y máximo dado. Se comparan imágenes de aplicaciones reales y se obtienen resultados interesantes.

## 1. Introducción

Como sabemos, una imagen se puede representar como un arreglo de 3 matrices numéricas de enteros, una por cada canal de color RGB o según sea su espacio de colores. Por ello se pueden hacer uso de las operaciones matemáticas para procesar imágenes. Podemos encontrar las operaciones aritméticas sencillas para matrices incluida la división. Esto es posible ya que se entiende que es la operación componente a componente, los cuales son números reales operando entre ellos.

También podemos manejar las matrices como funciones, conjuntos y espacios vectoriales. Los conjuntos con sus clásicas operaciones: conjunción, disyunción, adición y complemento.

Estas operaciones podrían parecer una simple extensión de las matemáticas ya descubiertas a lo largo de los tiempos, pero en aplicaciones médicas, biológicas, astronomía y geografía, una resta, negación o transformación podría resultar en una imagen más fácil de analizar.[1][2][3]

## 2. Metodología

Este trabajo se desarrollo en Google Colab. En este ambiente se puede compilar el lenguaje Python con una basta cantidad de librerías ya incluidas sin necesidad de instalarlas. Se utilizaron las librerías cv2 (OpenCV), numpy y matplotlib.pyplot. La primera sirve para manejo de imágenes como objetos que tiene diversos métodos definidos. Numpy es una librería optimizada para operaciones de arreglos (vectores, matrices, etc.), muy útil para reducir el código por sus métodos y la velocidad de

las operaciones. Por último, `matplotlib.pyplot` es una función para graficar, funciona muy similar al `plot` de Matlab.

1.- Para la imagen *Fig0303(a).tif* se realizarán 4 diferentes variantes. La primera será una negación de la imagen, la segunda un ajuste de intensidades entre 0.5 a .75, la tercera una transformación Gamma con  $\gamma = 2$  y la cuarta una transformación Gamma con  $\gamma = 0.5$ . Por último, comparar y responder ¿Cuál de las transformaciones anteriores le parece a usted una buena transformación para visualizar más fácilmente la lesión tumoral?

2.- Con la imagen *Fig0306(a).tif* se tendrán que utilizar las funciones del paso 1 y ver cuál tuvo el mejor resultado para observar el esqueleto.

3.- Con la misma imagen *Fig0306(a).tif*, primero será transformada y luego binarizada con valores 0 – 255 y reportar la que combinación con el mejor resultado. ¿Se queda satisfecho con el resultado obtenido? ¿Por qué?

4.- Definir una función llamada `ImEscala` que reciba de parámetros una imagen, un valor mínimo y uno máximo. La imagen será la operada y los valores serán los que acoten la escala que todos los valores de la matriz pertenecerán dentro del rango.

5.- Por último, usar la función definida en el paso anterior con la imagen *Fig0310(a).tif* primero con los valores 0 y 225, luego con 0 y 1 y después con el valor mínimo y máximo de la imagen. Comparar y discutir los resultados.

### 3. Resultados

1.- Las imágenes resultantes son los que se muestran en la Figura 1.

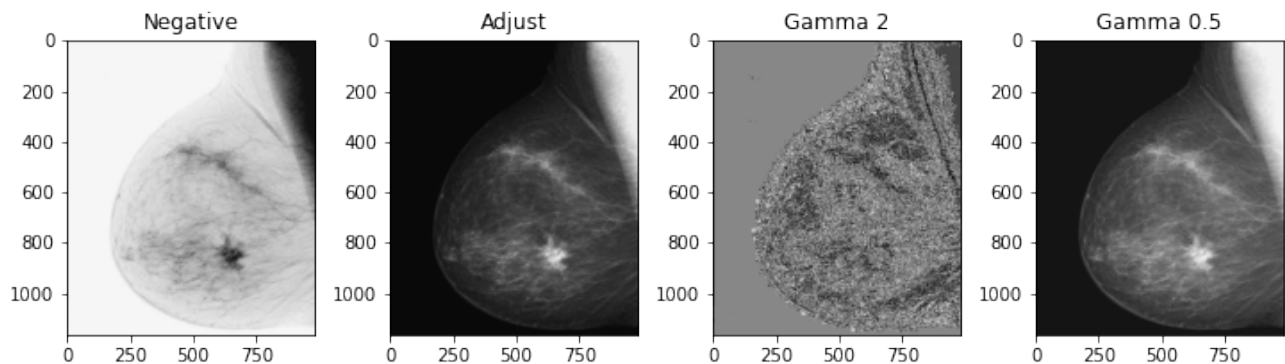


Figura 1: Comparación de la Negación, Ajuste y Transformación Gamma con la imagen de una mama.

La imagen que utilizaría para analizar la lesión sería la negación.

2.- Las imágenes resultantes para una imagen del esqueleto humano son los que se muestran en la Figura 2.

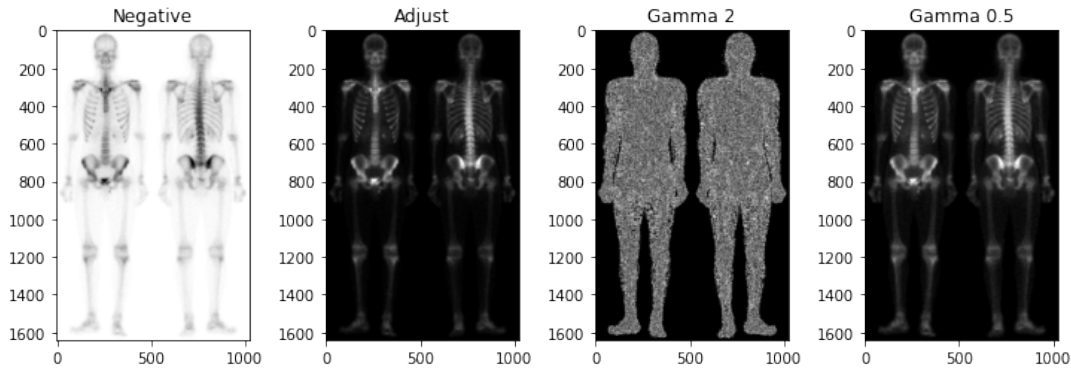


Figura 2: *Comparación de la Negación, Ajuste y Transformación Gamma con la imagen del esqueleto humano.*

La mejor imagen del esqueleto es la Negación y con  $\text{Gamma} = 0.5$ .

3.- Las transformaciones fueron: La Negación menos Gamma 3 y menos el Ajuste. Por último, se binariza a valores solo de 0 ó 255. El resultado se muestra en la Figura 3.

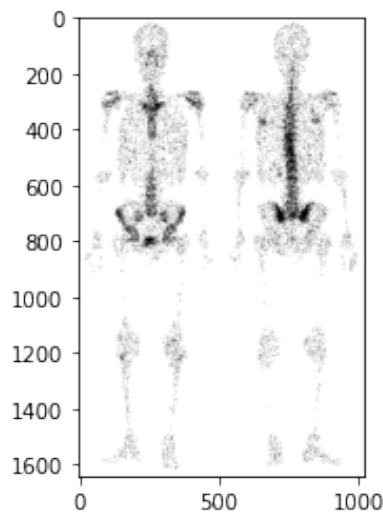


Figura 3: *Resultado de la combinación de transformaciones.*

El resultado fue el mejor que se pudo encontrar pero no considero que sea el mejor esperado.

4.- La función *ImEscala* quedó como:

```
def ImEscala(img, minf, maxf):

    x_min = img.min()
    x_max = img.max()

    img = minf + ((img - x_min) / (x_max - x_min)) * (maxf - minf)

    return img
```

5.- Las imágenes resultantes de las escalas de la imagen de un asteroide en la Figura 4.

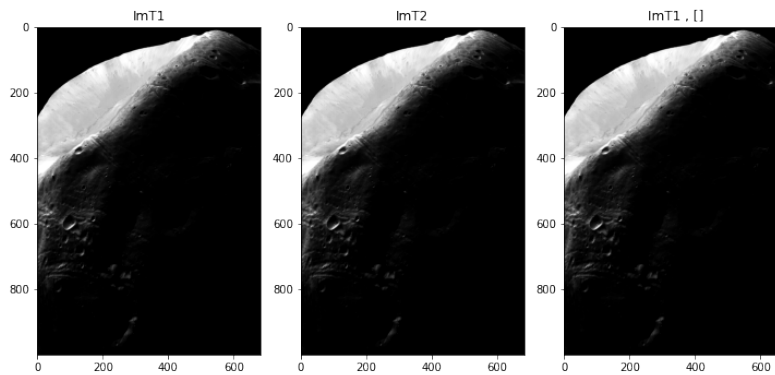


Figura 4: Resultado de la escala  $[0,255]$ ,  $[0,1]$  y la escala normal.

## 4. Discusiones

1.- De las 4 imágenes, la transformación Gamma 2 es la menos visible para el propósito de resaltar la lesión tumoral. La transformación Gamma 0.5 resalta en general el seno y el ajuste resalta más la lesión. Estas dos tienen como base el color negro y por tanto contienen menos valores en sus matrices. La negación tiene una base más blanca y resalta más la lesión con un color más notable y oscuro. En mi opinión, esta resulta más útil para vislumbrar la lesión.

2.- Los filtros en esta imagen actúan de forma similar a la anterior. La imagen Gamma 2 es la menos distinguible y parece completar la imagen del cuerpo humano. El Ajuste y la Gamma 0.5 son en base oscura y la imagen clara. La Negación es al contrario y la transformación Gamma 0.5 es más distinguible al Ajuste. Y la Negación y la Gamma 0.5 son las mejores para ver la figura.

3.- La mejor combinación fue la mostrada en los resultados, operaciones aritméticas con matrices y una transformación para la binarización. Como se puede apreciar, la imagen Gamma 2 fue útil para lograr esta imagen, ya que otras combinaciones complicaban el resultado al momento de binarizar. Esto es por el umbral configurado en la función que se utilizó en Python. Se esperaba un

mejor resultado pero fue el mejor que se pudo encontrar.

4.- La función sigue una ecuación por la cual, cuando la componente es el mínimo de la imagen, la fracción se anula y el resultado es el mínimo que fue pedido. Cuando la componente es el máximo, la fracción es una unidad y el producto resultante resta el valor independiente y el resultado es el máximo que fue pedido.

5.- Las imágenes son muy parecidas, no parecen tener un cambio significativo. Las escalas se cumplen, los valores en las matrices se encuentran en sus respectivos rangos y con valores flotantes. La última de las 3, pareciera ser la primera, ya que el mínimo es 0 y el máximo es 255. Por lo que, el resultado tendría que ser el mismo matemáticamente.

## 5. Conclusión

Las transformaciones Gamma tiene buenos resultados entre 1.9 y 0.4 pero a valores grandes se distorsiona mucho la imagen. La Negación es muy útil para destacar componentes de una imagen. El Ajuste tiene buenos resultados. Las operaciones aritméticas también son muy útiles para obtener mejores imágenes, en algunas aplicaciones, una simple operación da el punto de partida para un análisis.

Las escalas no varían los resultados, hablando de un mínimo global y un máximo global, por supuesto que si entre esos rangos están los valores, en ese caso la imagen será diferente.

Recurrir a operaciones simples, podría ser la solución a un problema aparentemente complejo.

## A. Código

El código diseñado para esta práctica se encuentra en un repositorio de Github para su visualización y con la opción de abrir en Google Colab.

## Referencias

- [1] Chung, B. W. (2017). *Getting Started with Processing and OpenCV*, pages 1–37. Apress, Berkeley, CA.
- [2] Solem, J. (2012). *Programming Computer Vision with Python: Tools and algorithms for analyzing images*, pages 73–273. O'Reilly Media.
- [3] Woods, G. . (2018). *Digital Image Processing*, pages 1–40. Pearson, New York, NY.