



**INSTITUTO POLITECNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO**

“Practica 1-Imagenes”

-Tinoco Videgaray Sergio Ernesto

Grupo: 4BV1

Materia: Procesamiento digital de imágenes

INSTITUTO POLITÉCNICO NACIONAL



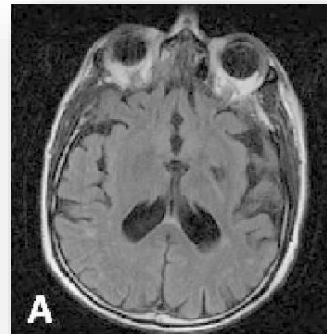
ESCOM®

09/03/22

En el área de la computación y de la inteligencia artificial existe un campo muy específico que se encarga de la captura y procesamiento de las imágenes de manera digital, por medio del cual podemos contrastar y estudiar múltiples elementos dentro de una imagen y con el uso de diferentes algoritmos podemos determinar que elementos se encuentran y que características tienen.

En el desarrollo de esta practica se va a realizar un pequeño algoritmo que nos permitirá abrir una imagen, importarla a una matriz y finalmente aplicar un algoritmo para generar una nueva imagen que nos muestre el negativo de la imagen original.

Para este caso vamos a trabajar con una tomografía que fue obtenida por medio de una resonancia magnética:



- Desarrollo:

Para el algoritmo de esta practica se va a trabajar con el lenguaje de programación Python.

Primeramente, se van a importar las bibliotecas correspondientes:

De la biblioteca PILLOW se va a importar el paquete Image que nos va a permitir abrir un archivo de imagen de la computadora.

Se va a importar la Biblioteca Numpy asignándole como alias “np”.

De igual forma se va a importar el paquete Pyplot de la biblioteca matplotlib.

```
from PIL import Image
import numpy as np
from matplotlib import pyplot as plt
```

Posteriormente se va a abrir el archivo de imagen de la tomografía:

```
img=Image.open("Documents/ProgramasPython/Practicas/PDI/RM/RM-A.jpg")
```

Posteriormente se convertirá la imagen en escala de grises para trabajar únicamente con los valores de grises.

```
img=img.convert("L")
```

Una vez convertida la imagen, se carga en una matriz usando la función "asarray" del paquete Numpy

```
matriz=np.asarray(img)
```

Se declara una segunda matriz para la imagen negativa del tamaño de la imagen original:

```
matrizInv=[[ 0 for i in range(matriz.shape[1])] for j in range(matriz.shape[0])]
```

Se recorre la matriz de la imagen original para asignarle el valor $[L-1]-r$ a la matriz de la imagen invertida.

```
i=0
while(i<357):
    j=0
    while(j<344):
        matrizInv[i][j]=255-matriz[i][j]
        j+=1
    i+=1
```

Carga los datos de la matriz original en la función imshow para posteriormente generar el grafico y mostrarlo en pantalla.

```
plt.imshow(matriz,cmap="gray")
plt.title("Imagen original")
plt.show
```

Finalmente carga la segunda matriz y genera el grafico correspondiente.

```
plt.imshow(matrizInv,cmap="gray")  
plt.title("Imagen invertida")  
plt.show()
```

- Resultados:

