INSTITUTO POLITECNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO

“Practica 5-Operaciones aritméticas y lógicas entre imágenes”

-Tinoco Videgaray Sergio Ernesto

Grupo: 4BV1

Materia: Procesamiento digital de imágenes

08/04/22

Algunas de las operaciones entre imágenes nos permiten unir dos imágenes en una sola o bien resaltar algún elemento dentro de la imagen.

En esta práctica se van a aplicar las operaciones lógicas y aritméticas entre dos imágenes en blanco y negro.

- Desarrollo:

Para el algoritmo de esta práctica 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”.

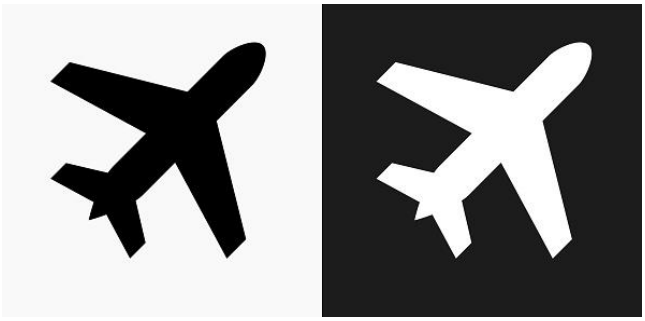
De igual forma se va a importar el paquete Pyplot de la biblioteca matplot.

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

Operaciones aritméticas:

Suma:

Para el caso de la suma se van a trabajar con 2 imágenes en blanco y negro del contorno de un avión.



Se cargan las imágenes en el programa y se carga dentro de sus matrices correspondientes.

```
imgA=Image.open("P5/imgA.jpg").convert("L")
imgB=Image.open("P5/imgB.jpg").convert("L")
matrizA=np.asarray(imgA)
matrizB=np.asarray(imgB)
```

Se obtienen los valores de ancho y alto de la matriz

```
alto=matrizA.shape[0]
ancho=matrizA.shape[1]
```

De declara una tercera matriz para almacenar los valores de la suma:

```
matrizC = [ [ 0 for i in range(ancho) ] for j in range(alto) ]
```

Se recorren ambas matrices con los índices i, j realizando la suma para cada par de pixeles en la posición i, j dentro de sus matrices.

Dentro del ciclo while anidado se realiza la suma y se divide entre 2 para obtener el valor de salida.

```
i=0
while(i<alto):
    j=0
    while(j<ancho):
        matrizC[i][j]=(matrizA[i][j]+matrizB[i][j])//2
        j+=1
    i+=1
```

Finalmente se cargan los datos de la matriz en la función “imshow” y se muestran en un gráfico al usuario.

```
    plt.imshow(matrizC,cmap="gray")
    plt.title("Suma")
  plt.show()
```

De manera similar se va a realizar la resta de ambas matrices indicando que, si el valor obtenido resulta ser negativo, se asigne el valor de 0.

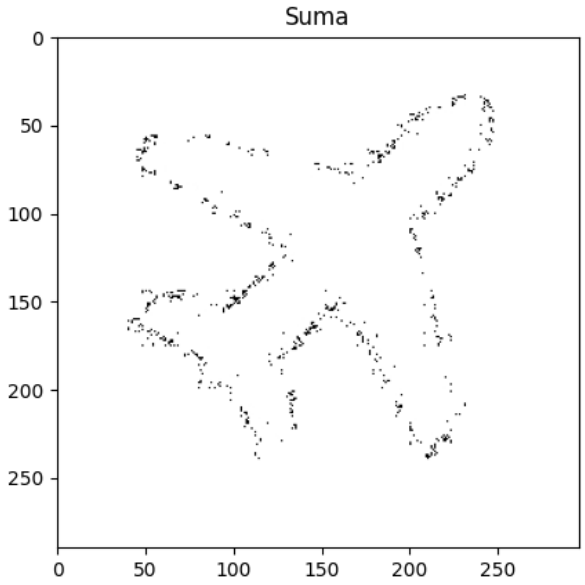
```
i=0
while(i<alto):
    j=0
    while(j<ancho):
        matrizC[i][j]=(matrizA[i][j]-matrizB[i][j])
        if(matrizC[i][j]<0):
            matrizC[i][j]=0
        j+=1
    i+=1
```

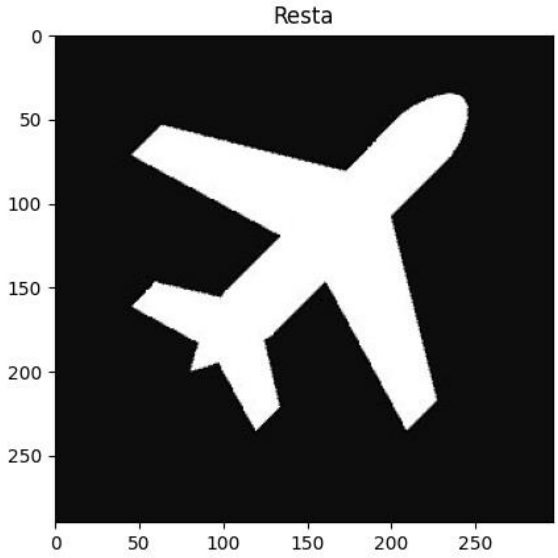
Se muestra la imagen generada en un gráfico.

```
plt.imshow(matrizC,cmap="gray")
```

```
plt.title("Resta")
plt.show()
```

- Resultados:





Ahora se van a realizar las operaciones lógicas entre 2 imágenes.

Operación AND:

Para la operación AND se va a trabajar con 2 imágenes en blanco y negro de 2 aviones cazas militares.

Se cargan los datos de las imágenes en el programa y se cargan los valores dentro de 2 matrices en niveles de gris.

```
imgA=Image.open("P5/cA.jpg").convert("L")
imgB=Image.open("P5/cB.jpg").convert("L")
matrizA=np.asarray(imgA)
matrizB=np.asarray(imgB)
```

Se obtienen los valores de alto y ancho de la imagen a trabajar.

```
alto=matrizA.shape[0]
ancho=matrizA.shape[1]
```

Se declara una tercera matriz para los valores generados.

```
matrizC = [ [ 0 for i in range(ancho) ] for j in range(alto) ]
```

Se recorren ambas matrices por medio de un ciclo while anidado.

Por medio del operador '&' se realiza la operación AND bit a bit con el valor de nivel de gris de ambas imágenes.

```
i=0
while(i<alto):
    j=0
    while(j<ancho):
        matrizC[i][j]=matrizA[i][j]&matrizB[i][j]
        j+=1
    i+=1
```

Se muestra la imagen generada en un gráfico.

```
plt.imshow(matrizC,cmap="gray")
plt.title("AND")
plt.show()
```

- Resultado:

Operación OR:

Para las operaciones OR y XOR se hará uso de una imagen de un avión caza de

1950 y una plantilla con un recuadro negro para resaltar la parte de la cabina.

Se cargan las imágenes dentro de dos matrices A y B.

```
imgA=Image.open("P5/gA.jpg").convert("L")
imgB=Image.open("P5/gB.jpg").convert("L")

matrizA=np.asarray(imgA)
matrizB=np.asarray(imgB)
```

Se obtienen los valores de alto y ancho de la imagen a trabajar.

```
alto=matrizA.shape[0]
ancho=matrizA.shape[1]
```

Se declara una tercera matriz para los valores generados.

```
matrizC = [ [ 0 for i in range(ancho) ] for j in range(alto) ]
```

Se recorren ambas matrices por medio de un ciclo while anidado.
Por medio del operador '|' se realiza la operación OR bit a bit con el valor de nivel de gris de ambas imágenes.

```
i=0
while(i<alto):
    j=0
    while(j<ancho):
        matrizC[i][j]=matrizA[i][j]|matrizB[i][j]
        j+=1
    i+=1
```

Se muestra la imagen generada en un gráfico por medio de la función show.

```
plt.imshow(matrizC,cmap="gray")
```

```
plt.title("OR")
plt.show()
```

Operación XOR:

Utilizando las mismas imágenes de la operación OR se realiza la operación de OR exclusiva o “XOR” por medio del operador ‘^’.

```
i=0
while(i<alto):
    j=0
    while(j<ancho):
        matrizC[i][j]=matrizA[i][j]^matrizB[i][j]
        j+=1
    i+=1
```

Finalmente se cargan los datos de la matriz C en un grafico generado por la función imshow y se muestran por medio de la función show.

```
plt.imshow(matrizC,cmap="gray")
plt.title("XOR")
plt.show()
```

- Resultados: