PROCESAMIENTO DIGITAL DE IMÁGENES

PAO 2023-1

Estudiante: David Bravo Serna

Actividad:

LAB 8

Contents

Tabla de imágenes

Ejercicio 1

Implementar los algoritmos de realzado

1. Alargamiento
2. Compresión
3. Desplazamiento.

Imagen original

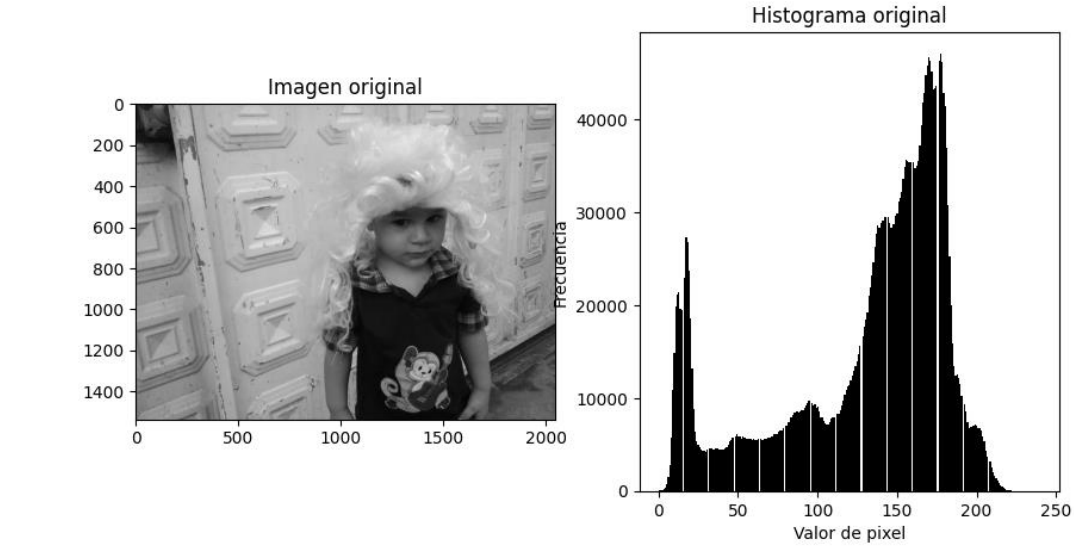

Ilustración 1Imagen original ejercicio 1

Los valores contenidos en esta imagen van desde el 0 al 240

Alargamiento

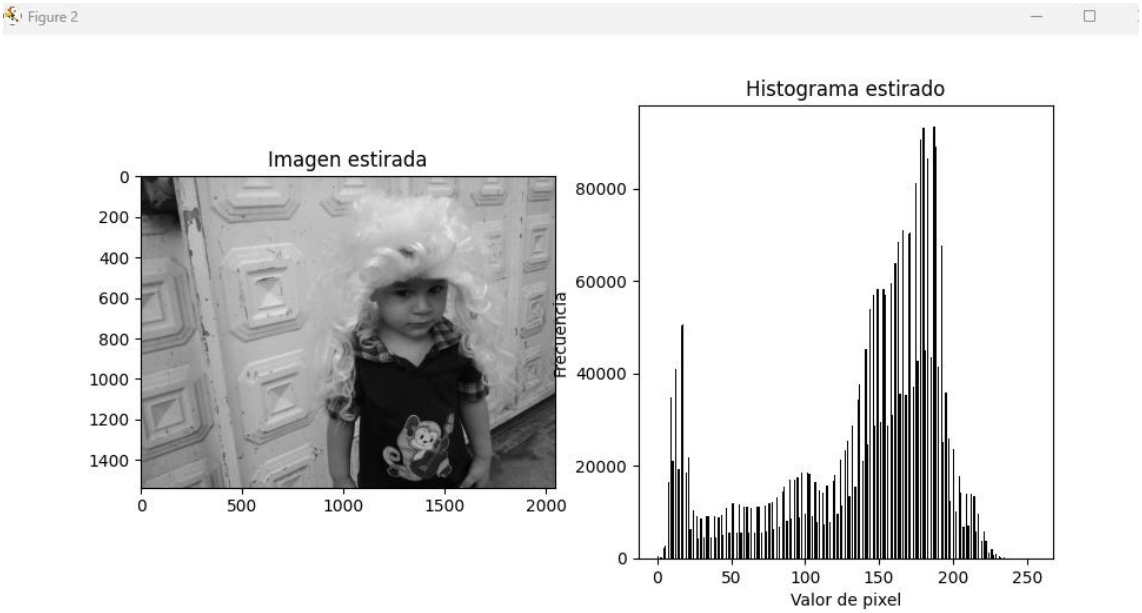

Ilustración 2 imagen alargada

```python
def histogram_stretch(image, MIN=0, MAX=255):
    # Alargamiento de histogramas (stretch)
    image = image.astype(float)
    I_MIN = np.min(image.ravel())
    I_MAX = np.max(image.ravel())
    print (I_MIN)
    print (I_MAX)
    image = (image - I_MIN) / (I_MAX - I_MIN) # normalización al rango [0, 1]
    stretched = image * (MAX - MIN) + MIN  # estiramiento al rango [MIN, MAX]
    return stretched.astype(np.uint8)
```

Ilustración 3 Código del método alargamiento

Para el alargamiento estamos utilizando un estrechamiento de 50 – 200 y luego aplicamos un alargamiento

```
stretched = histogram_stretch(histogram_shrink(image, 50, 200))
```

Comprimido

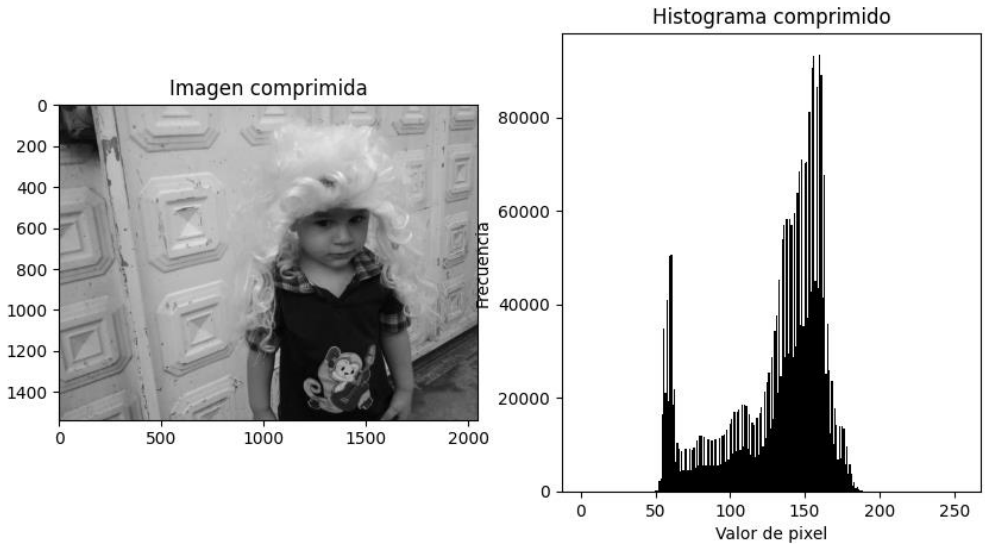

Ilustración 4 imagen compresa

```
def histogram_shrink(image, Shrink_MIN, Shrink_MAX):
    # Compresión de histogramas (shrink)

# Shrink_MIN y Shrink_MAX son los límites del intervalo de compresión (0 <= Sh
rink_MIN < Shrink_MAX <= 255)
    image = image.astype(float)
    I_MIN = np.min(image.ravel())
    I_MAX = np.max(image.ravel())
    shrunk = Shrink_MIN + ((Shrink_MAX - Shrink_MIN) / (I_MAX - I_MIN)) * (
image - I_MIN)
    return shrunk.astype(np.uint8)
```

Ilustración 5 código del método comprimir

Para la compresión estamos utilizando unos valores idénticos al alargamiento

```
shrunk = histogram_shrink(image, 50, 200)
```

Desplazamiento

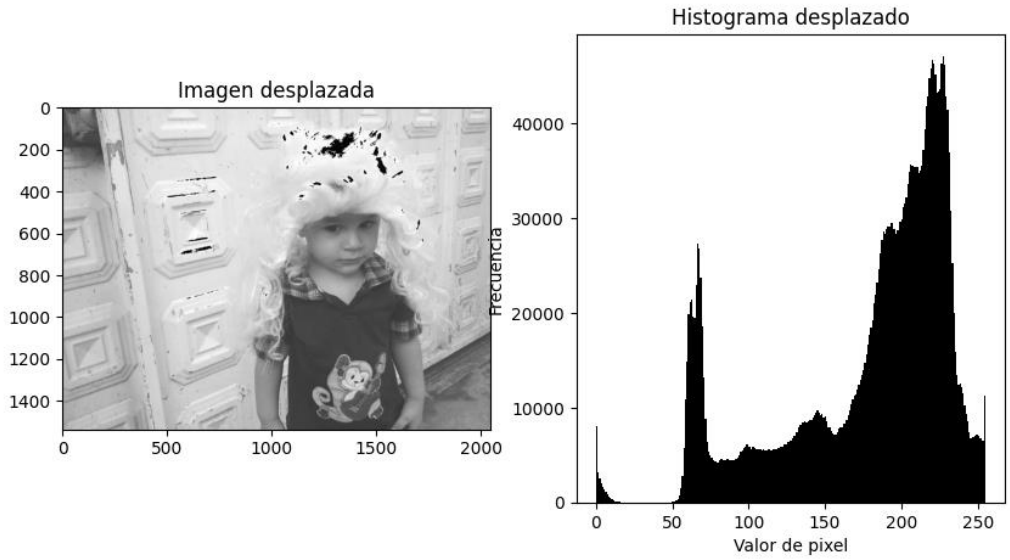

Ilustración 6 imagen desplazada

```python
def histogram_slide(image, d):
    # Desplazamiento de histogramas (slide)
    # d es la cantidad de desplazamiento (puede ser positiva o negativa)
    slid = image + d
    return np.clip(slid, 0, 255).astype(np.uint8)
```

Ilustración 7 código del método desplazamiento

Para el desplazamiento estamos utilizando un valor igual a 50

```python
slid = histogram_slide(image, 50)
```

Ejercicio 2

Imagen original

Ilustración 8 imagen original usada en el ejercicio 2

```
img = cv2.imread('example3.jpg', cv2.IMREAD_GRAYSCALE)

fft_img = np.fft.fft2(img)
img_fftshift = np.fft.fftshift(fft_img)
```

Ilustración 9 código donde cargamos la imagen

Cargamos la imagen y la pasamos al dominio frecuencial utilizando la transformada de Fourier

Ilustración 10 dominio frecuencial con un corte de frecuencia de 40

En esta imagen podemos ver la frecuencia de corte que es igual a 40, esto nos ayudara aplicando el filtro paso bajo

Ilustración 11 imagen resultante con el corte frecuencial

Imagen luego de aplicarle el filtro, nos damos cuenta de que tiene un efecto similar a un efecto gaussiano

```
def filtro_pb(shape,cut_freq):
    rows,cols = shape
    center_r,center_c = rows//2, cols//2
    filter = np.zeros((rows,cols))
    filter[center_r- cut_freq:center_r + cut_freq,center_c -cut_freq:center_c +
cut_freq]=1
    return filter

fshift = filtro_pb(img_fftshift.shape,40) * img_fftshift
f_ishift = np.fft.ifftshift(fshift)
img_back = np.fft.ifft2(f_ishift)
img_back = np.abs(img_back)
img_back_frecuen = cv2.normalize(img_back, None, 0, 255, cv2.NORM_MINMAX)
```

Ilustración 12 código donde se aplica el filtro de paso bajo

Ilustración 13 mascara que contiene los bordes provenientes del estiramiento

En esta imagen podemos ver solo los contornos de la imagen, esto nos ayudara para poder reparar la imagen aplicando este efecto o filtro y repara la imagen afectada por el filtro paso bajo

```
imgCompress = np.clip(img-img_back_frecuen,0,255)
stretched = histogram_stretch(imgCompress)
result = np.add(img,stretched)
```

Ilustración 14 código donde se hace el estiramiento

Ilustración 15 imagen resultante

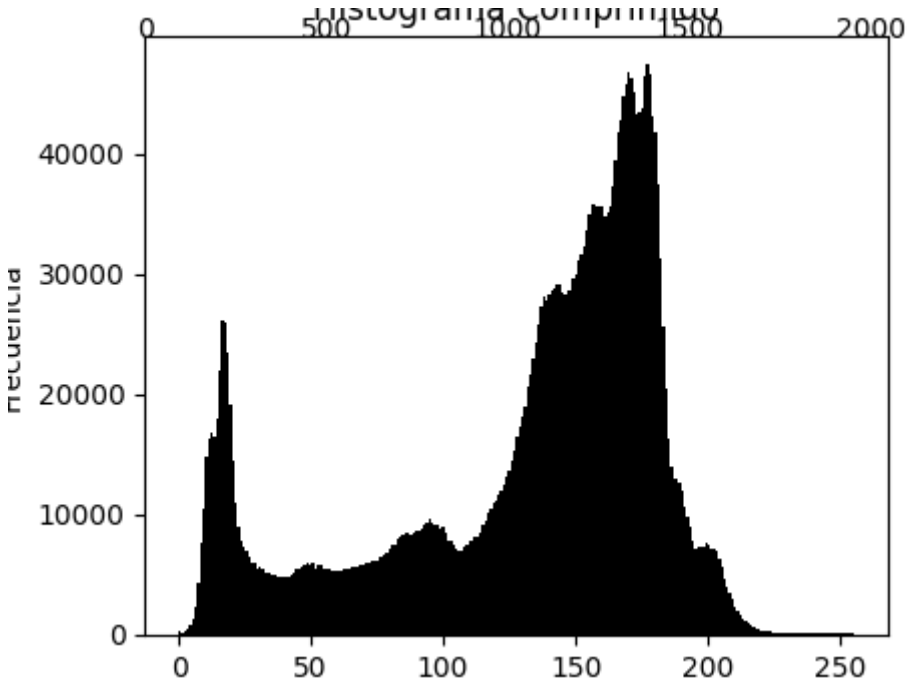

Ilustración 16 histograma de la imagen resultante

Podemos visualizar los resultamos exitosos que con ayuda de la manipulación de histograma podemos reparar imágenes con una alta calidad