



**Instituto Politécnico Nacional
Escuela Superior de Cómputo**



**Asignatura
Image Analysis**

**Alumno
Carapia González José Ricardo
García Medina Saúl
Guadarrama Hidalgo Luis Jorge**

**Título
Práctica 1 reporte técnico. Ajuste de brillo**

**Entrega
Día 25/04/2021**

Introducción	2
Objetivo del proyecto	4
Marco teórico	4
Desplazamiento del histograma	4
Contracción del histograma	4
Expansión del histograma	4
Ecuilización del histograma	5
Desarrollo - Propuesta	6
Cargar imagen	6
Histogramas (color y grises)	6
Ecuilizar imagen	7
Expandir histograma	7
Contraer histograma	8
Desplazar histograma	9
Ecuilización exponencial	10
Resultados	10
Desplazamiento del histograma.	10
Contracción del histograma.	11
Expansión del histograma.	11
Ecuilización exponencial.	12
Conclusión	12
Conclusión	12

1. Introducción

En la actualidad los medios gráficos de distribución de trabajos científicos están de la mano con la tecnología. Más en concreto, al precio de implantación de dicha tecnología.

La implementación del análisis de imágenes es muy importante dentro de la tecnología para resolver necesidades a un usuario, como también apoyar el avance en distintas profesiones y servicios, en esta materia existen técnicas para el manejo e interpretación de las imágenes, pero

¿Qué es una imagen ?

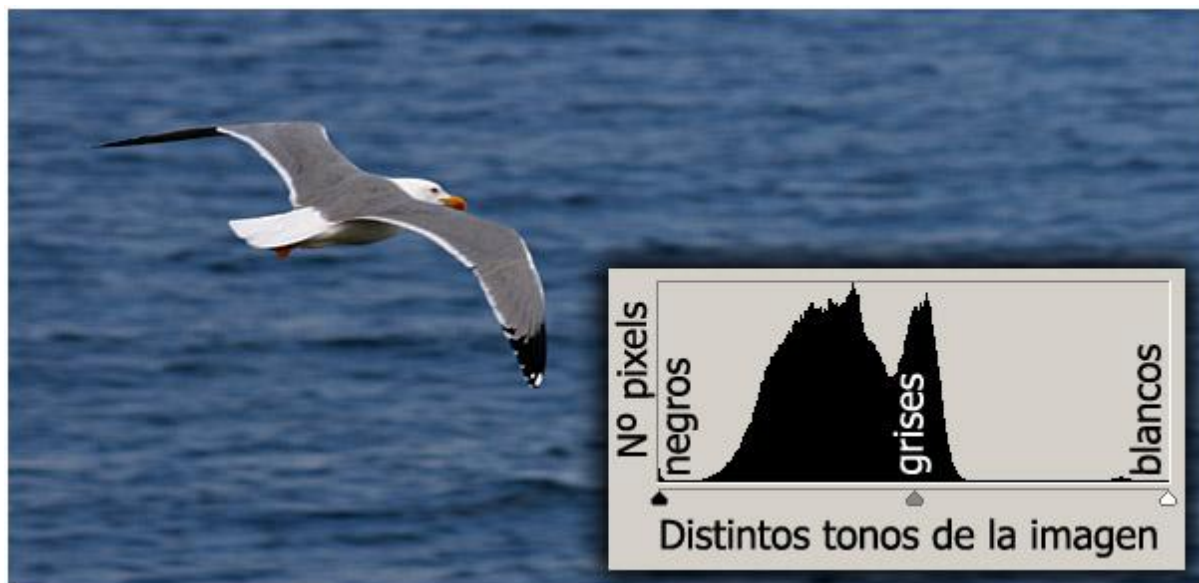
Una imagen puede ser definida como una función de dos dimensiones $f(x,y)$ donde x y y son las coordenadas espaciales (plano) y la amplitud de la función f en algún par de coordenadas (x,y) es llamada intensidad o nivel de gris de la imagen en ese punto. Cuando x , y y los valores de la amplitud de la función f son cantidades discretas finitas, a dicha imagen se le llama imagen digital.

La representación más simple de una imagen es una colección de puntos en un arreglo bidimensional, donde para cada punto se almacena una serie de parámetros propios de la imagen. Prácticamente para el registro de una imagen se usan formatos que están referenciados por el número de bits que son utilizados para la cuantización de la cantidad de luz recibida.

La mayor parte de los dispositivos simples permiten adquirir la imagen con una profundidad de 8 bits, los cuales permiten representar 256 niveles de gris, el rango típico va de $[0, 255]$. Cada coordenada en el mapa de bits se llama Pixel.

Las imágenes a color consisten regularmente de una composición de tres imágenes separadas en escala de grises, cada una representa los tres colores primarios; rojo, verde y azul; comúnmente conocido como RGB.

Y por último tenemos el histograma que es una representación gráfica de la distribución de los distintos tonos de una imagen. Puede ayudarnos para controlar la exposición en nuestras fotos, así como para corregir los colores.



El eje horizontal representa los diferentes tonos de gris desde el negro puro (a la izquierda) al blanco puro (a la derecha).

El eje vertical representa el número de píxeles que contiene la imagen para cada tono representado en el eje horizontal.

2. Objetivo del proyecto

Aproximar al estudiante a adquirir la habilidad de identificar y escoger la técnica a utilizar en un imagen en sus distintas presentaciones y distinguir todo tipo de técnicas básicas del histograma para mejorar las imágenes a trabajar.

3. Marco teórico

3.1. Desplazamiento del histograma

El desplazamiento de imagen aditivo se define bajo la forma $g(n)=[f(n)]+L$, donde L puede tomar cualquier valor dentro de la escala de grises asumiendo que las imágenes se han cuantificado en enteros en el rango $\{0\dots,K-1\}$. Por lo tanto, el histograma juega una relación dada por $H_g(k)=H_f(k-L)$, donde el valor de L corresponde al desplazamiento del histograma a la izquierda o a la derecha en la escala de grises. En la Figura 1, podemos ver el efecto del desplazamiento aditivo con $L=\pm 70$ y sus respectivos histogramas. Es evidente como las imágenes son afectadas a medida que desplazamos los histogramas hacia la izquierda o derecha en la escala de grises. A medida que el histograma es desplazado hacia la izquierda, la imagen se oscurece, por el contrario, a medida que es desplazado hacia la derecha, se esclarece. Estos efectos pueden ayudar a encontrar detalles o información en las imágenes las cuales a simple vista no son perceptibles.

3.2. Contracción del histograma

Es útil para reducir el contraste de los ng presentes en la imagen, hace una disminución del brillo. La contracción está dada por la siguiente expresión matemática:

$$g(i,j) = \left[\frac{C_{MAX} - C_{MIN}}{f(i,j)_{MAX} - f(i,j)_{MIN}} \right] [f(i,j) - f(i,j)_{MIN}] + C_{MIN}$$

Figura 1. Modelo matemático para la contracción del histograma

Donde:

$f(i,j)_{max}$ es el mayor nivel de gris de la imagen de entrada f .

$f(i,j)_{min}$ es el menor nivel de gris de la imagen de entrada f .

C_{max} y C_{min} corresponden al máximo y mínimo valor deseados en el resultado de la compresión del histograma.

3.3. Expansión del histograma

Este tipo de técnica, también llamada estiramiento de contraste es una operación de punto lineal que expande el histograma de la imagen para llenar todo el rango de escala de grises disponible. Un aspecto interesante de esta técnica, es que hoy en día, casi todas las cámaras de video comerciales de uso doméstico y profesionales aplican el estiramiento del histograma a escala completa como una función básica antes de que las imágenes sean incorporadas en la memoria. Está dado por la siguiente expresión matemática:

$$g(i, j) = \left[\frac{f(i, j) - f(i, j)_{\text{MIN}}}{f(i, j)_{\text{MAX}} - f(i, j)_{\text{MIN}}} \right] [MAX - MIN] + MIN$$

Figura 2. Modelo matemático para la expansión del histograma

Donde:

$f(i, j)_{\text{max}}$ es el mayor nivel de gris de la imagen de entrada f .

$f(i, j)_{\text{min}}$ es el menor nivel de gris de la imagen de entrada f .

MAX y MIN corresponden al máximo y mínimo valor deseados en el resultado de la compresión del histograma.

3.4. Ecualización del histograma

Una de las operaciones de punto no lineales más importantes es la ecualización por histogramas, también llamada aplanamiento de histograma. La idea subyace en una extensión del estiramiento de contraste visto en el punto anterior, en la cual no solo se llena el rango completo de escala de grises, sino que también lo distribuye de una manera uniforme en todo el rango. Existen distintos tipos de ecualización y son las siguientes (expresadas en su forma matemática):

Tabla 1. Tipos de ecualización según su distribución

Distribuciones	Expresiones
<i>Uniforme</i>	$F(g) = [g_{max} - g_{min}] P_g(g) + g_{min}$
<i>Exponencial</i>	$F(g) = g_{min} - \frac{1}{\alpha} \ln[1 - P_g(g)]$
<i>Rayleigh</i>	$F(g) = g_{min} + \left[2\alpha^2 \ln \left\{ \frac{1}{1 - P_g(g)} \right\} \right]^{\frac{1}{2}}$
<i>Hipercúbica</i>	$F(g) = \left(\left[\sqrt[3]{g_{max}} - \sqrt[3]{g_{min}} \right] P_g(g) + \sqrt[3]{g_{min}} \right)^3$
<i>Logaritmo hiperbólica</i>	$F(g) = g_{min} \left[\frac{g_{max}}{g_{min}} \right] P_g(g)$

con $P_g(g) = \sum_{g=0}^g p(g)$ y α un parámetro a definir previamente a la realización del ecualizado.

Figura 3. Modelo matemático de $P_g(g)$.

4. Desarrollo - Propuesta

Este apartado muestra todo el desarrollo de la práctica de la parte de software, mostrando su código y resultados.

Cargar imagen

```
def cargarImg(event):
    global file
    global filestatus
    global ruta
    global filename
    file = askopenfilename()
    filestatus["text"] = str(file)#int()""ddsd""
    ruta = str(file)
    filename = os.path.split(ruta)#F:/users/imagen.jpg
    #('F:/users/', 'imagen.jpg')
```

Histogramas (color y grises)

```

def histogramColor(event):
    if(ruta == None):
        messagebox.showerror("Error", "Selecciona una imagen primero")
    else:
        histograma.histogramaColor(filename[1])

def histogramGris(event):
    if(ruta == None):
        messagebox.showerror("Error", "Selecciona una imagen primero")
    else:
        histograma.histogramaGris(filename[1])

```

```

def histogramaColor(self, ruta):
    img = cv2.imread(ruta)
    cv2.imshow(ruta, img)

    color = ('b','g','r')

    for i, c in enumerate(color):
        hist = cv2.calcHist([img], [i], None, [256], [0, 256])
        plt.plot(hist, color = c)
        plt.xlim([0,256])

    plt.show()

    cv2.destroyAllWindows()

def histogramaGris(self, ruta):
    img = cv2.imread(ruta, cv2.IMREAD_GRAYSCALE)
    cv2.imshow(ruta, img)

    hist = cv2.calcHist([img], [0], None, [256], [0, 256])
    plt.plot(hist, color='gray' )

    plt.xlabel('intensidad de iluminacion')
    plt.ylabel('cantidad de pixeles')
    plt.show()

    cv2.destroyAllWindows()

```

Ecualizar imagen

```

def ecualizarImg(event):
    if(ruta == None):
        messagebox.showerror("Error", "Selecciona una imagen primero")
    else:
        histograma.ecualizacion(filename[1])

```

```
def ecualizacion(self, ruta):
    img = cv2.imread(ruta, cv2.IMREAD_GRAYSCALE)
    img = cv2.equalizeHist(img)

    cv2.imshow('Histogramas', img)
    hist = cv2.calcHist([img], [0], None, [256], [0, 256])
    plt.plot(hist, color='gray' )

    plt.xlabel('intensidad de iluminacion')
    plt.ylabel('cantidad de pixeles')
    plt.show()
    cv2.waitKey()
```

Expandir histograma

```
def expandirHist(event):
    if(ruta == None):
        messagebox.showerror("Error", "Selecciona una imagen primero")
    else:
        intervaloMin = simpledialog.askinteger(title="Nuevo intervalo mínimo", prompt="Cuál será el nuevo intervalo mínimo para el histograma")
        intervaloMax = simpledialog.askinteger(title="Nuevo intervalo máximo", prompt="Cuál será el nuevo intervalo máximo para el histograma")
        histograma.expansion(filename[1], intervaloMin, intervaloMax)
```

```
def expansion(self, ruta, nuevoMin, nuevoMax):
    img = cv2.imread(ruta, cv2.IMREAD_GRAYSCALE)
    tamañoImg = img.shape #(ancho, alto)
    print(tamañoImg)
    pixeles = [] #lista
    for x in range(tamañoImg[0]): #(min, max)
        for y in range(tamañoImg[1]):
            pixeles.append(img.item(x, y))
    npPixeles = np.array(pixeles)
    infoImg = np.unique(npPixeles) #arreglo
    max = np.amax(infoImg)
    min = np.amin(infoImg)
    recta = Mates()
    for x in range(tamañoImg[0]):
        for y in range(tamañoImg[1]):
            img.itemset((x, y), recta.ecRecta(min, max, nuevoMin, nuevoMax, img.item(x, y)))
    cv2.imshow("Histograma", img)
    hist = cv2.calcHist([img], [0], None, [256], [0, 256])
    plt.plot(hist, color='gray' )

    plt.xlabel('intensidad de iluminacion')
    plt.ylabel('cantidad de pixeles')
    plt.show()
    cv2.destroyAllWindows()
```

```
def ecRecta(self, x1, x2, y1, y2, intensidadPixel):
    #Recordemos que una recta es  $y = mx + b$ 
    m = (y2-y1)/(x2-x1)
    b = y1 - (m * x1)

    return m * intensidadPixel + b
```


Contraer histograma

```
def contraerHist(event):
    if(ruta == None):
        messagebox.showerror("Error", "Selecciona una imagen primero")
    else:
        intervaloMin = simpledialog.askinteger(title="Nuevo intervalo mínimo", prompt="Cuál será el nuevo intervalo mínimo para el histograma?")
        intervaloMax = simpledialog.askinteger(title="Nuevo intervalo máximo", prompt="Cuál será el nuevo intervalo máximo para el histograma?")
        histograma.contraccion(filename[1], intervaloMin, intervaloMax)
```

```
def contraccion(self, ruta, nuevoMin, nuevoMax):
    img = cv2.imread(ruta, cv2.IMREAD_GRAYSCALE)
    tamañoImg = img.shape
    pixeles = []
    for x in range(tamañoImg[0]):
        for y in range(tamañoImg[1]):
            pixeles.append(img.item(x, y))
    npPixeles = np.array(pixeles)
    infoImg = np.unique(npPixeles)#arreglo
    max = np.amax(infoImg)
    min = np.amin(infoImg)
    contraccion = Mates()
    for x in range(tamañoImg[0]):
        for y in range(tamañoImg[1]):
            img.itemset((x, y), contraccion.contraer(nuevoMax, nuevoMin, max, min, img.item(x, y)))
    cv2.imshow("Histograma", img)
    hist = cv2.calcHist([img], [0], None, [256], [0, 256])
    plt.plot(hist, color='gray' )

    plt.xlabel('intensidad de iluminacion')
    plt.ylabel('cantidad de pixeles')
    plt.show()
    cv2.destroyAllWindows()
```

```
def contraer(self, cmax, cmin, rmax, rmin, rk):
    return ((cmax - cmin) / (rmax - rmin)) * (rk - rmin) + cmin
```

Desplazar histograma

```
def desplazarHist(event):
    if(ruta == None):
        messagebox.showerror("Error", "Selecciona una imagen primero")
    else:
        des = simpledialog.askinteger(title="Desplazar histograma", prompt="Cuál será el desplazamiento del histograma?")
        histograma.desplazamiento(filename[1], des)
```

```
def desplazamiento(self, ruta, des):
    img = cv2.imread(ruta, cv2.IMREAD_GRAYSCALE)
    tamañoImg = img.shape
    desplazamiento = Mates()
    for x in range(tamañoImg[0]):
        for y in range(tamañoImg[1]):
            img.itemset((x, y), desplazamiento.desplazar(des, img.item(x, y)))
    cv2.imshow("Histograma", img)
    hist = cv2.calcHist([img], [0], None, [256], [0, 256])
    plt.plot(hist, color='gray' )

    plt.xlabel('intensidad de iluminacion')
    plt.ylabel('cantidad de pixeles')
    plt.show()
    cv2.destroyAllWindows()
```

```
def desplazar(self, desplazamiento, pixel):
    return pixel + desplazamiento
```

Ecualización exponencial

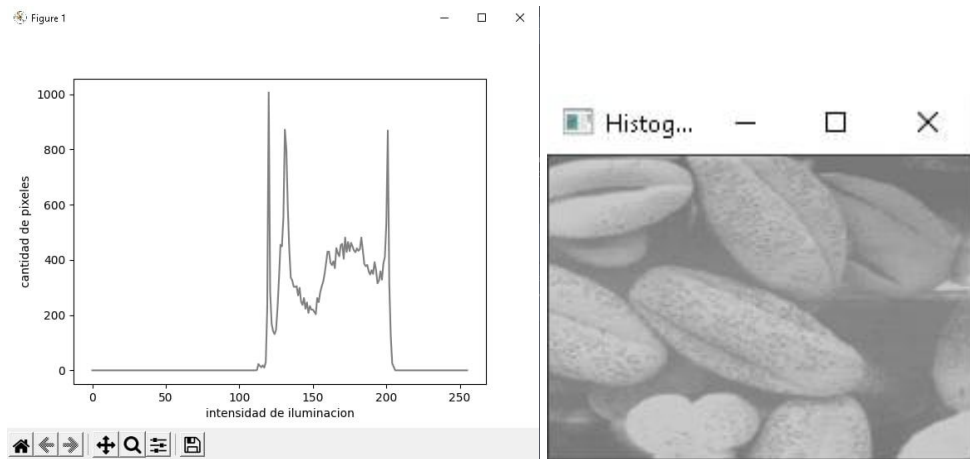
```
def exponencial(event):
    if(ruta == None):
        messagebox.showerror("Error", "Selecciona una imagen primero")
    else:
        alfa = simpledialog.askfloat(title="Define el alfa", prompt="Cuál será el valor alfa?:")
        histograma.ecExp(filename[1], alfa)
```

```
def ecExp(self, ruta, alfa):
    img = cv2.imread(ruta, cv2.IMREAD_GRAYSCALE)
    tamañoImg = img.shape
    totalElementos = img.size
    pixeles = []
    for x in range(tamañoImg[0]):
        for y in range(tamañoImg[1]):
            pixeles.append(img.item(x, y))
    npPixeles = np.array(pixeles)
    infoImg = np.unique(npPixeles, return_counts=True) #Tupla (arreglo, num apariciones)
    frecuencias = np.array(infoImg[1])
    probabilidad = []
    min = np.amin(infoImg[0])
    for i in range(0, len(infoImg[0])):
        probabilidad.append(frecuencias[i]/totalElementos)
    ecu = Mates()
    pG = []
    for i in range(0, len(probabilidad) - 1):
        if(i == 0):
            pG.append(probabilidad[i])
        else:
            pG.append(probabilidad[i] + pG[i-1])
    sk = []
    cont = 0
    for i in range(0, len(probabilidad)-1):
        sk.append(ecu.ecualizacionExp(min, alfa, pG[i], npPixeles[i]))
    print(pixeles)
    c = 0
    for index, value in enumerate(pixeles):
        if value == pixeles[index]:
            if c < 93:
                pixeles[index] = sk[c]
            else:
                break
            c = c + 1
    caca = []
    for x in range(tamañoImg[0]):
        for y in range(tamañoImg[1]):
            img.itemset((x, y), pixeles[cont])
            caca.append(pixeles[cont])
            cont = cont + 1
    print(len(pixeles))
```

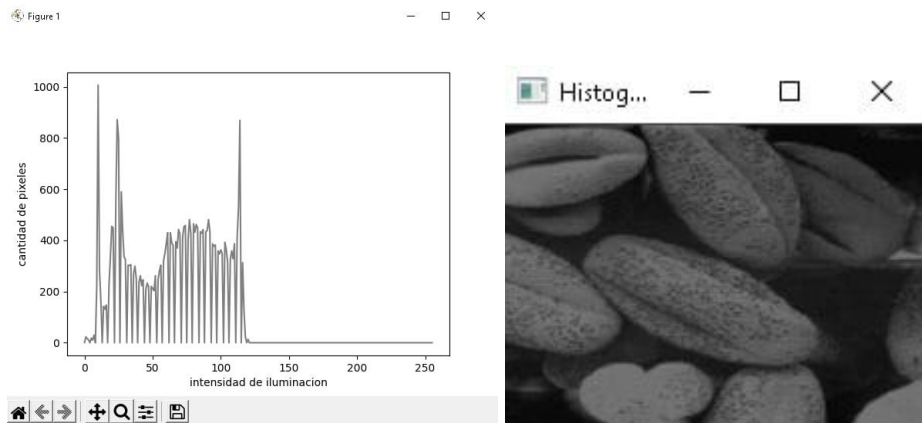
```
def ecualizacionExp(self, rmin, alfa, probabilidades, rj):
    return round(rmin - ((1/alfa) * log(1 - probabilidades)))
```

5. Resultados

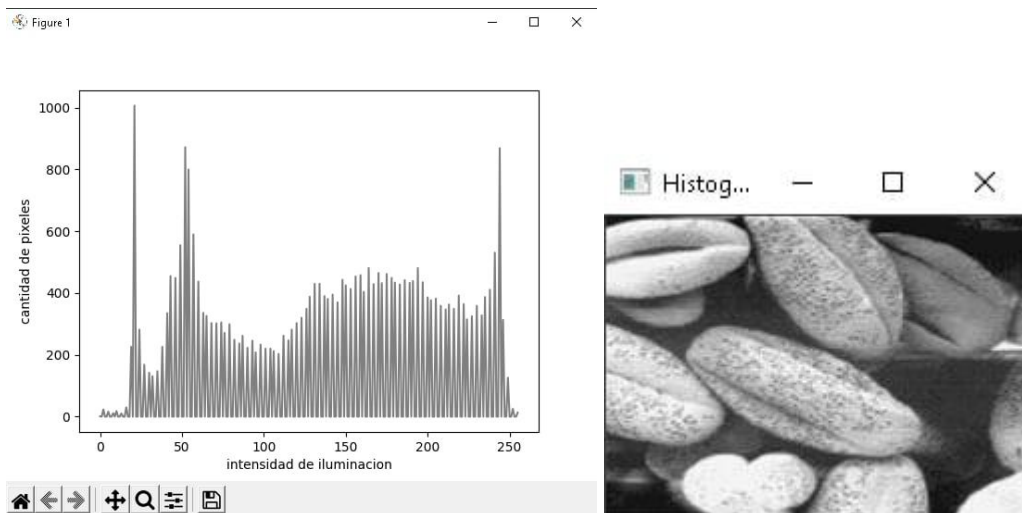
5.1. Desplazamiento del histograma.



5.2. Contracción del histograma.

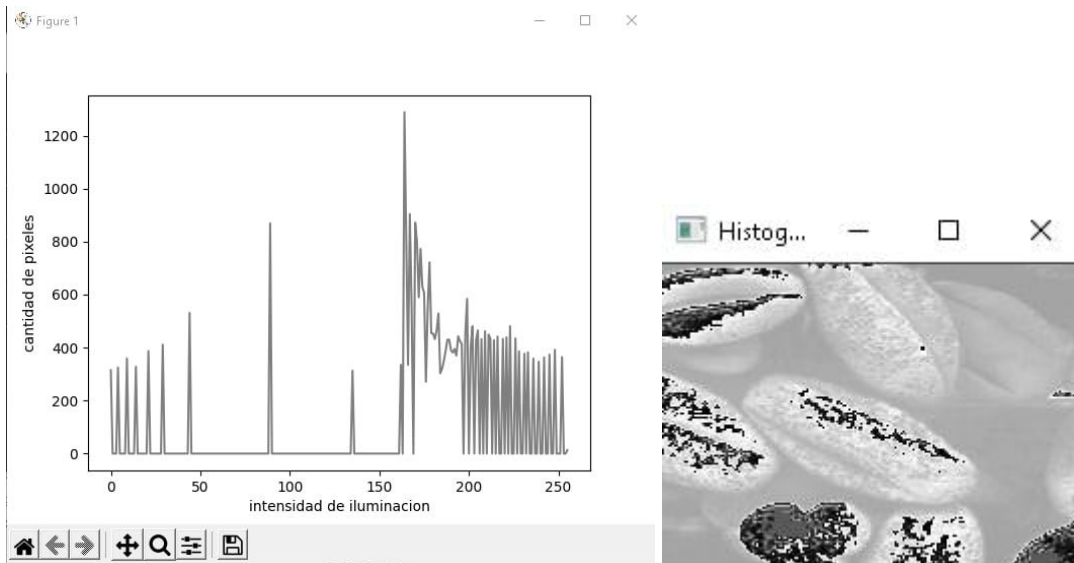


5.3. Expansión del histograma.



5.4. Ecualización exponencial.

Alfa=0.023



5.5. Conclusión

Como se puede observar la mejor forma de tratar el problema de esta imagen es usando la expansión del histograma para poder ver claramente los granos de arroz y diferenciarlos

6. Conclusión

Podemos concluir que el ajuste de brillo es una herramienta que nos ayuda a operar sobre el histograma de una imagen y así hacer que ésta tenga un mejor brillo y contraste. Con las técnicas vistas anteriormente pudimos observar de manera más técnica cómo operar con los píxeles de una imagen matemáticamente para mejorar el brillo de la misma. Una cosa importante a resaltar es que la contracción y la expansión del histograma es exactamente el mismo procedimiento, de hecho si nos fijamos en sus modelos matemáticos nos encontraremos que están planteados de la misma manera y que lo único que cambia es la forma en la que llaman a cada una de sus variables, pero en esencia son lo mismo, solo que en la contracción se va a delimitar un rango menor al rango de la expansión de la imagen, puesto que la contracción busca oscurecer la imagen, mientras que la expansión busca aumentar el rango de niveles de gris. En cuanto a la ecualización, cada una ofrece distintos resultados y ya es decisión del usuario usar la que más le convenga, en nuestro caso que nos tocó la ecualización exponencial, observamos en la práctica que ésta oscurece muchísimo la imagen, por lo que puede ser útil para imágenes muy brillantes. El desplazamiento del histograma es la operación más sencilla de realizar puesto que solo suma o resta el valor deseado a cada uno de los píxeles sin mayor complejidad. Para

concluir, con esta práctica nos dimos cuenta de lo poderoso que es el histograma como herramienta, pues proporciona mucha información útil acerca de la imagen, la cual da pauta para decidir qué operación es la más conveniente de realizar para mejorar la imagen.