TRATAMIENTO DE SOMBRAS EN FOTOGRAFIAS DE DOCUMENTOS

```
# importamos librerias utilizadas
import cv2
import numpy as np
import matplotlib.pyplot as plt
from scipy import ndimage, misc
import matplotlib.pyplot as plt
# definimos unafuncion para graficar las imagenes
def imprimir pic(pic,etiqueta):
    plt.figure()
    plt.imshow(pic, cmap = 'gray')
    plt.title(etiqueta)
    plt.axis('off')
#imagen entrada = "cejem1"
#input pic = cv2.imread(imagen entrada+".png")
#cv2.imwrite(imagen entrada+".jpeg", input pic)
Seleccionamos la imagen con la cual se va a trabajar
imagen entrada = "input3"
input pic = cv2.imread(imagen entrada+".jpeg")
METODO BASE
Generamos la mascara o "perfil de sombra"
# Paquetes necesarios para la morfología matemática
from skimage.morphology import erosion, dilation, opening, closing
# Elementos estructurales
from skimage.morphology import disk, diamond, ball, rectangle, star
from scipy import ndimage as ndi
#reducimos la pic a grises
input pic = cv2.cvtColor(input pic, cv2.COLOR BGR2GRAY)
```

#realizamos una copia de la pic recibida para poder modificarla en un

#las sisguientes 3 lineas se encargar de normalizar la pic de 0 a 255

espacio de memoria diferente
mask = np.copy(input pic)

mask = mask - mask.min()
mask = mask/mask.max()

mask = mask*255

```
#Creamos la mascara con un filtro de mediana, consideramos las letras
como ruido
for i in range(3):
    mask = ndimage.median filter(mask, size=18,mode='reflect')
mask gris = np.copy(mask)
imprimir pic(mask, 'Máscara en grises')
#la maskara debe estar en blanco y negro y no en escala de grises, asi
que se realiza un if:
promedio = np.mean(mask)
for (cor_y,cor_x), value in np.ndenumerate(mask):
    if mask[cor_y,cor_x]promedio: #Good Pixel
        mask[cor y, cor x] = 0.1
    elif mask[cor_y,cor_x]>promedio: #Bad Pixel
        mask[cor y, cor x] = 254.9
imprimir pic(mask, 'Máscara')
imprimir_pic(input_pic, 'original')
```

Máscara en grises



Máscara



original

```
Y = 0 o X = 3, Y = 0 o X = 4, Y = 0 o X = 5, Y = 0. Por lo tamto P(U = 0) = 0.28. Y = 0.0 X = 3, Y = 0.0 X = 4, Y = 0.0 X = 5, Y = 0. Por lo tamto P(U = 0) = 0.28. Y = 0.0 X = 3, Y = 0.0 X = 4, Y = 0.0 X = 5, Y = 0. Por lo tamto P(U = 0) = 0.28. Y = 0.0 X = 3, Y = 0.0 X = 4, Y = 0.0 X = 5, Y = 0. Por lo tamto P(U = 0) = 0.28. Y = 0.0 X = 3.0 X = 0.0 X = 3.0 X = 0.0 X = 0
```

Definimos la funcion de contraste que apartir de un percentil permite

imprimir pic(zona oscura, "zona oscura")

```
def funcion de contraste(pic,percentil):
    limite = np.percentile(pic,percentil)
    blancos = pic >= limite
    pic = pic + 255*blancos
    negros saturados = pic<limite</pre>
    blancos saturados = pic>=limite
    pic[negros saturados] = 0
    pic[blancos saturados] = 255
    return pic
Definimos el array "Zona Oscura"
zona oscura = np.full(input pic.shape,np.nan)
limite = np.nanmean(input pic)
for (coor_y,coor_x), value in np.ndenumerate(input_pic):
    aux = mask[coor y:coor y+1,coor x:coor x+1]
    if aux<limite: #Good Pixel</pre>
        zona_oscura[coor_y,coor_x] = input_pic[coor_y,coor_x]
    elif aux>limite: #Bad Pixel
```

zona oscura

```
distribution problems as the continues of the continues o
```

Definimos el array "Zona Clara"

```
zona_clara = np.full(input_pic.shape,np.nan)
limite = np.nanmean(input_pic)

for (coor_y,coor_x), value in np.ndenumerate(input_pic):
    aux = mask[coor_y:coor_y+1,coor_x:coor_x+1]
    if aux>limite: #Good Pixel
        zona_clara[coor_y,coor_x] = input_pic[coor_y,coor_x]
    elif aux<=limite: #Bad Pixel
        None

imprimir_pic(zona_clara,"zona_clara")</pre>
```

zona_clara



Definimos la funcion que nos permite conocer el punto optimo para contrastar la imagen def contraste_optimo_por_MSE(segmento,pic):

```
output = 255 - np.nan_to_num(segmento,nan=0)
base = 255 -pic
errorlist = []
for i in range(30):

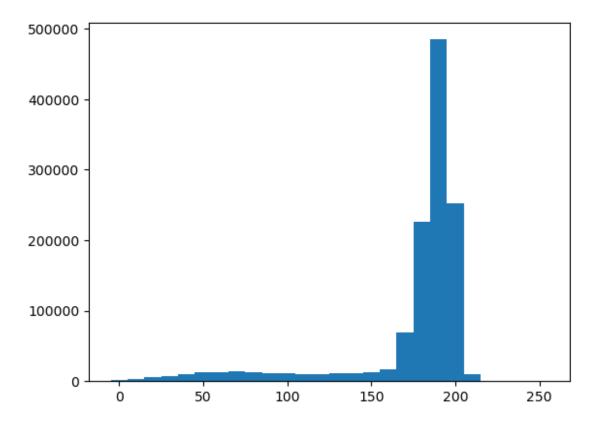
    f = funcion_de_contraste(output,i)
    error = np.sum((base-f)**2)
    errorlist.append(error)

m = np.arange(30)
#plt.scatter( m,errorlist)
#plt.title("Error vs intensidad (por percentile)")
#plt.show()
return (m[np.argmin(errorlist)]+1)
```

METODO BASE

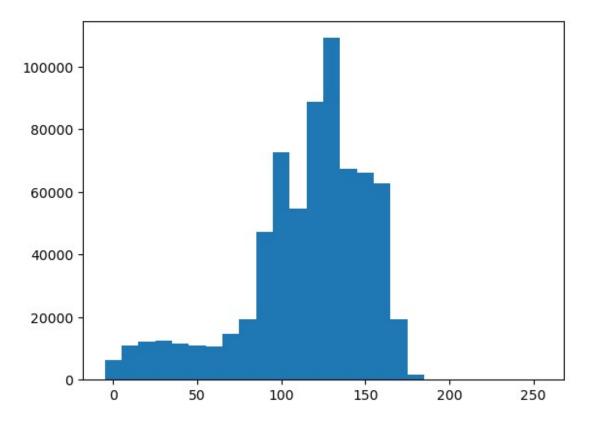
Obtenemos el histograma de la zona clara con el fin de obtener su moda

```
a,b = np.histogram(zona_clara,bins=np.arange(0,270,10)) plt.bar( b[0:26],a[0:26],width=10) moda_zona_clara = b[np.argmax(a)+1]
```



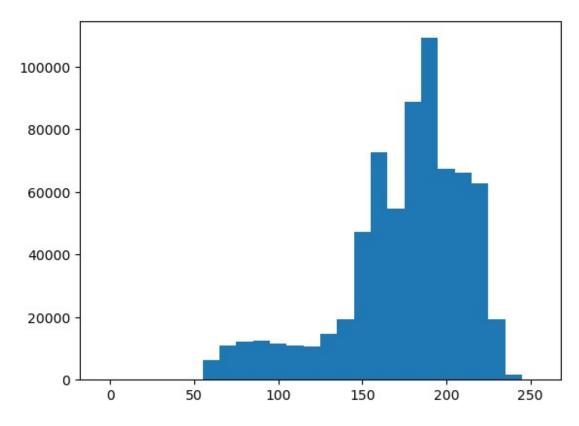
Obtenemos el histograma de la zona oscura con el fin de obtener su moda

```
a,b = np.histogram(zona_oscura,bins=np.arange(0,270,10))
plt.bar( b[0:26],a[0:26],width=10)
moda_zona_oscura = b[np.argmax(a)+1]
```



```
Modificamos el histograma de la zona oscura con el fin de igualar la moda de ambas zonas
osc = np.copy(zona_oscura) + ( -moda_zona_oscura + moda_zona_clara)
#osc = zona_oscura
lessThen0 = osc<0
moreThen255 = osc>255
osc[lessThen0] = 0
osc[moreThen255] = 255
#imprimir_pic(osc, "osc")
a,b = np.histogram(osc,bins=np.arange(0,270,10))
plt.bar(b[0:26],a[0:26],width=10)
#moda_zona_oscura = b[np.argmax(a)+1]
```

<BarContainer object of 26 artists>



imprimir_pic(osc,"Zona oscura con histograma modificado")

Zona oscura con histograma modificado



imprimir_pic(zona_clara, "zona_clara")

zona_clara

```
Funcionari de una variable alestoria 100

Y = 0 o X = 3, Y = 0 o X = 4, Y = 0 o X = 5, Y = 0. Por lo tanto P(U = 0) = 0.28.

El resto de las probabilidades asociadas con U se pueden obtener de una manera sentimiente por tanto, la distribución de probabilidades de U se puede resumir sentimiente por tanto, la distribución de probabilidades de las variables akentorias V y W como se definió anteriormente se subbladera de las variables akentorias V y W como se definió anteriormente se subbladera de las variables bidimensional continua y si Z = H<sub>1</sub>(X, Y) es una S(1), X, Y) es una variable bidimensional continua y si Z = H<sub>1</sub>(X, Y) es una variable bidimensional continua y si Z = H<sub>1</sub>(X, Y) es una variable de encontrar va sida pe sa " o n" i con "cado. Para p y discultir la la problema necesitamos un teorema que vi si Z = H<sub>2</sub>(X, Y). Para resolver interior la figuración de la viva de la "río, bo" se Z = H<sub>2</sub>(X, Y). Para resolver interior la figuración de la viva desempeña solo un papel di minima. Con el fin de encontrar rema 6.3.

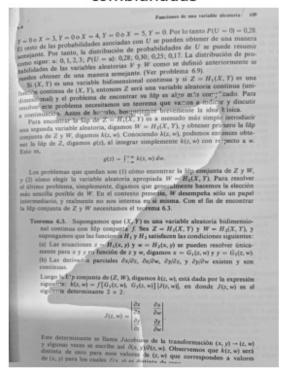
Y es una variable aleatoria bidimensional de la viva de la la viva de la parciales de la "río, de la viva de la contección de la viva digamos x eficia, vi) y w = G(1,x, vi) y = G(1,x, vi) z = G(1,x, vi) y = G(1,
```

Combinamos ambas zonas: la zona clara y la zona oscura modificada

```
z1 = np.nan_to_num(zona_clara,0)
z2 = np.nan_to_num(osc,0)

suma_parcial = (z1 + z2)
imprimir_pic(suma_parcial,"combianadas")
```

combianadas



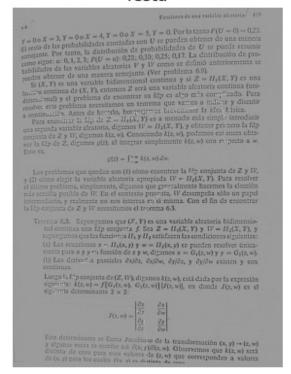
Restamos las mascara (en tonos de gris) de la imagen original

```
resta_parcial = np.copy(suma_parcial)
mask_gris = ndimage.median_filter(suma_parcial,
size=15,mode='reflect')

resta_parcial = ((suma_parcial - mask_gris)+255)/(2)
lessThen0 = resta_parcial<0
moreThen255 = resta_parcial>255
resta_parcial[lessThen0] = 0
resta_parcial[moreThen255] = 255

imprimir_pic(resta_parcial, "resta")
```

resta



Aplicamos un contraste optmizado por MSE a la imagen de "resta parcial"

```
p_indicado = contraste_optimo_por_MSE(255 - resta_parcial,255 -
input_pic)
final_metodo_BASE = funcion_de_contraste(resta_parcial,p_indicado)
imprimir_pic(final_metodo_BASE,"final_metodo_BASE")
```

final metodo_BASE

Y=00 X=3, Y=00 X=4, Y=00 X=5, Y=0. Por lo tanto P(U=0)=0.28. El esto de las probabilidades asociadas con U as pueden obtener de una manera smejante. Por tanto, la distribución de probabilidades de U se puede resumir como signe: u:0,1,2,3; P(U=u):0,28:(0,30:(0,25;0,17.La) distribución de probabilidades de las variables aleatorias <math>V y W como se definió anteriormente se poden obtener de una manera semejante. (Ver problema 6.9). Si (X,Y) es una variable bidissensional continua y si $Z=H_1(X,Y)$ es una función continua q si $Z=H_1(X,Y)$ es una función continua Q si $Z=H_1(X,Y)$ es una variable aleatoria coetínua (unidiresional) y el problema de encontrar su foly es algo más complicado. Para renovirar la file $Z=H_1(X,Y)$ es a menudo más simple instrodución. Para coetínua (in $Z=H_1(X,Y)$) es a menudo más simple instrodución una segunda variable aleatoria, digamos $W=H_2(X,Y)$, y obtener primero la foly conjunta de Z y W, digamos $Z=H_1(X,Y)$, al integrar simplemente $Z=H_1(X,Y)$ con respecto $Z=H_1(X,Y)$.

$$g(z) = \int_{-\infty}^{+\infty} k(z, w) \, dw.$$

Los problemas que quedan son (I) cômo encontrar la fide conjunta de Z y W_c y (2) cômo elegir la variable aleatoria apropiada $W=H_2(X,Y)$. Para resolver el útimo problema, simplemente, digamos que generalmente hacemos la elección raís sencifia posible de W. En el contexto presente, W desempeña sólo un papel intermediario, y realmente no nos interesa en sí misma. Con el fin de encontrar la fisp conjunta de Z y W necesitamos el teorema 6.3.

Terretta 6.3. Supongamos que (X,Y) es una variable aleatoria bidimensional continua con fdp conjunta f. Sea $Z=H_1(X,Y)$ y $W=H_2(X,Y)$, y supongamos que las funciones H_1 y H_2 satisfacen las condiciones siguientes: (a) Las ecuaciones $z=H_1(x,y)$ y $w=H_2(x,y)$ se pueden resolver ûnicamente para x y y en función de z y w, digames $x=G_1(z,w)$ y $y=G_2(z,w)$. (b) Las derivadas parciales $\partial_x(\partial z,\ \partial_x(\partial w,\ \partial_y(\partial z,\ y\ \partial_y)\partial w$ existen y son

Luego la ldp conjunta de (Z, W), digamos h(z, w), está dada por la expresión signiente: $h(z, w) = f[G_1(z, w), G_2(z, w)] [J(z, w)]$, en donde J(z, w) es el signiente determinante 2×2 :

$$J(z, w) = \begin{vmatrix} \frac{\partial x}{\partial z} & \frac{\partial x}{\partial w} \\ \frac{\partial y}{\partial z} & \frac{\partial y}{\partial w} \end{vmatrix}$$

Este determinante se llama Jacobiaso de la transformación $(x,y) \to (x,w)$ y algunas veces se escribe asi $\hat{c}(x,y)\hat{c}(x,w)$. Observemos que k(x,w) será dutistas de occo para esto valores de (x,w) que corresponden a valores de (x,y) para los cuales f(x,w) es dutintos de corresponden a valores

METODO B

```
# Importar las librearias necesarias:
```

```
import matplotlib.pyplot as plt
from PIL import Image
import scipy.ndimage as snd
import numpy as np
import cv2
from skimage.filters import threshold otsu
from cv2 import threshold, adaptiveThreshold
# Declarar metodos para aplicar Clausura y Threshold:
def closing(image array):
  # Estructurar elemento de clausura:
  huella = np.ones((40, 40))
  # Aplicar clausura de grises:
  fondo = snd.grey_closing(image_array, footprint=huella)
  # Se resta el fondo de la imagen:
  fondo libre = (image array.astype(np.float64) -
fondo.astype(np.float64))
  # Se reescala la imagen con el fondo libre de 0 a 255:
  denominador = (fondo libre.max() - fondo libre.min())
```

```
fondo libre normalizado = (fondo libre - fondo libre.min())*
255/denominador
  # Convertir fondo libre normalizado a uint8:
  fondo libre normalizado = fondo_libre_normalizado.astype(np.uint8)
  # Convertir fondo libre normalizado a imagen:
  fondo libre normalizado = Image.fromarray(fondo libre normalizado)
  return fondo libre normalizado
def threshold(image array):
  th = cv2.adaptiveThreshold(np.asarray(image array),
    255, # Maximo valor assignado al valor de un pixel que excede el
'threshold'.
    cv2.ADAPTIVE THRESH MEAN C, # suma ponderada "gaussiana" de los
vecinos.
    cv2.THRESH BINARY, # Tipo de threshold.
    15, # Tamaño del bloque (ventana 5x5)
    12)
  # Convertir th a imagen:
  th = Image.fromarray(th)
  return th
# Importar imagen:
input image = input pic
# Convertir la imagen en arreglo:
image array = np.asarray(input image)
# Aplicar Clausura:
closing image = closing(image array)
# Aplicar Threshold:
threshold image = threshold(closing image)
input image
array([[129, 126, 121, ..., 203, 203, 203],
       [130, 128, 124, ..., 203, 203, 203],
       [130, 130, 129, ..., 203, 203, 203],
       [103, 103, 104, \ldots, 191, 191, 191],
       [103, 103, 103, \ldots, 191, 191, 191],
       [103, 103, 103, ..., 191, 191, 191]], dtype=uint8)
closing image
```

Y = 0 o X = 3, Y = 0 o X = 4, Y = 0 o X = 5, Y = 0. Por lo tanto P(U = 0) = 0.28. El resto de las probabilidades asociadas con U se pueden obtener de una manera semejante. Por tanto, la distribución de probabilidades de U se puede resumir como sigue: u: 0, 1, 2, 3; P(U = u): 0.28; 0.30; 0.25; 0.17. La distribución de probabilidades de las variables aleatorias V y W como se definió anteriormente se pueden obtener de una manera semejante. (Ver problema 6.9).

Si (X, Y) es una variable bidimensional continua y si $Z = H_1(X, Y)$ es una función continua de (X, Y), entonces Z será una variable aleatoria continua (unidimensional) y el problema de encontrar su fdp es algo más complicado. Para resolver este problema necesitamos un teorema que vamos a indicar y discutir a continuación. Antes de hacerlo, bosquejemos brevemente la idea básica.

Para encontrar la fdp de $Z = H_1(X, Y)$ es a menudo más simple introducir una segunda variable aleatoria, digamos $W = H_2(X, Y)$, y obtener primero la fdp conjunta de Z y W, digamos k(z, w). Conociendo k(z, w), podemos entonces obtener la fdp de Z, digamos g(z), al integrar simplemente k(z, w) con respecto a w. Esto es,

$$g(z) = \int_{-\infty}^{+\infty} k(z, w) dw.$$

Los problemas que quedan son (1) cómo encontrar la fdp conjunta de Z y W, y (2) cómo elegir la variable aleatoria apropiada $W = H_2(X, Y)$. Para resolver el último problema, simplemente, digamos que generalmente hacemos la elección más sencilla posible de W. En el contexto presente, W desempeña sólo un papel intermediario, y realmente no nos interesa en sí misma. Con el fin de encontrar la sdp conjunta de Z y W necesitamos el teorema 6.3.

Teorema 6.3. Supongamos que (X, Y) es una variable aleatoria bidimensional continua con fdp conjunta f. Sea $Z = H_1(X, Y)$ y $W = H_2(X, Y)$, y supongamos que las funciones H_1 y H_2 satisfacen las condiciones siguientes: (a) Las ecuaciones $z = H_1(x, y)$ y $w = H_2(x, y)$ se pueden resolver únicamente para x y y en función de z y w, digamos $x = G_1(z, w)$ y $y = G_2(z, w)$. (b) Las derivadas parciales $\partial x/\partial z$, $\partial x/\partial w$, $\partial y/\partial z$, y $\partial y/\partial w$ existen y son continuas.

Luego la fdp conjunta de (Z, W), digamos k(z, w), está dada por la expresión signiente: $k(z, w) = f[G_1(z, w), G_2(z, w)] |J(z, w)|$, en donde J(z, w) es el siguiente determinante 2 × 2:

$$J(z, w) = \begin{vmatrix} \frac{\partial x}{\partial z} & \frac{\partial x}{\partial w} \\ \frac{\partial y}{\partial z} & \frac{\partial y}{\partial w} \end{vmatrix}$$

Este determinante se llama Jacobiano de la transformación $(x, y) \rightarrow (z, w)$ y algunas veces se escribe así $\partial(x, y)/\partial(z, w)$. Observemos que k(z, w) será distinta de cero para esos valores de (z, w) que corresponden a valores de (x, y) para los cuales f(x, v) es distinto de coro

 $Y = 0 \circ X = 3$, $Y = 0 \circ X = 4$, $Y = 0 \circ X = 5$, Y = 0. Por lo tanto P(U = 0) = 0.28. El resto de las probabilidades asociadas con U se pueden obtener de una manera semejante. Por tanto, la distribución de probabilidades de U se puede resumir como sigue: u: 0, 1, 2, 3; P(U = u): 0,28; 0,30; 0,25; 0,17. La distribución de probabilidades de las variables aleatorias V y W como se definió anteriormente se pueden obtener de una manera semejante. (Ver problema 6.9).

Si (X, Y) es una variable bidimensional continua y si $Z = H_1(X, Y)$ es una función continua de (X, Y), entonces Z será una variable aleatoria continua (unidimensional) y el problema de encontrar su sdp es algo más complicado. Para resolver este problema necesitamos un teorema que vamos a indicar y discutir a continuación. Antes de hacerlo, bosquejemos brevemente la idea básica.

Para encontrar la fdp de $Z = H_1(X, Y)$ es a menudo más simple introducir una segunda variable aleatoria, digamos $W = H_2(X, Y)$, y obtener primero la fdp conjunta de Z y W, digamos k(z, w). Conociendo k(z, w), podemos entonces obtener la sdp de Z, digamos g(z), al integrar simplemente k(z, w) con respecto a w. Esto es,

$$g(z) = \int_{-\infty}^{+\infty} k(z, w) dw.$$

Los problemas que quedan son (1) cómo encontrar la fdp conjunta de Z y W, y (2) cómo elegir la variable aleatoria apropiada $W = H_2(X, Y)$. Para resolver el último problema, simplemente, digamos que generalmente hacemos la elección más sencilla posible de W. En el contexto presente, W desempeña sólo un papel intermediario, y realmente no nos interesa en sí misma. Con el fin de encontrar la sdp conjunta de Z y W necesitamos el teorema 6.3.

Teorema 6.3. Supongamos que (X, Y) es una variable aleatoria bidimensional continua con fdp conjunta f. Sea $Z = H_1(X, Y)$ y $W = H_2(X, Y)$, y supongamos que las funciones H_1 y H_2 satisfacen las condiciones siguientes: (a) Las ecuaciones $z = H_1(x, y)$ y $w = H_2(x, y)$ se pueden resolver únicamente para x y y en función de z y w, digamos $x = G_1(z, w)$ y $y = G_2(z, w)$. (b) Las derivadas parciales $\partial x/\partial z$, $\partial x/\partial w$, $\partial y/\partial z$, y $\partial y/\partial w$ existen y son continuas.

Luego la fdp conjunta de (Z, W), digamos k(z, w), está dada por la expresión siguiente: $k(z, w) = f[G_1(z, w), G_2(z, w)] |J(z, w)|$, en donde J(z, w) es el siguiente determinante 2×2 :

$$J(z, w) = \begin{vmatrix} \frac{\partial x}{\partial z} & \frac{\partial x}{\partial w} \\ \frac{\partial y}{\partial z} & \frac{\partial y}{\partial w} \end{vmatrix}.$$

Este determinante se llama Jacobiano de la transformación $(x, y) \rightarrow (z, w)$ y algunas veces se escribe así $\partial(x,y)/\partial(z,w)$. Observemos que k(z,w) será distinta de cero para esos valores de (z, w) que corresponden a valores de (x, y) para los cuales f(x, v) es distinta de coro

METODO A

mask = dilation(image=mask, selem=disk(40))

```
imprimir_pic(mask, 'Máscara')
imprimir_pic(input_pic, 'original')
```

/tmp/ipykernel_6417/2346952556.py:1: FutureWarning: `selem` is a
deprecated argument name for `dilation`. It will be removed in version
1.0. Please use `footprint` instead.

mask = dilation(image=mask, selem=disk(40))

Máscara



original

```
Y = 0 o X = 3, Y = 0 o X = 4, Y = 0 o X = 5, Y = 0. Por lo tanto P(U = 0) = 0.28.  
El resto de las probabilidades asociadas con U se pueden obtener de una manera  
El resto de las probabilidades asociadas con U se pueden obtener de una manera  
El resto de las probabilidades asociadas con U se pueden obtener de una manera  
el resto de las virulados de distribución de probabilidades de U se puede resumir  
surrisante. Por tanto, la distribución de probabilidades de las variabble aleatorias V y W comos se definió anteriormente se  
habilidades de las variabble aleatorias V y W como se definió anteriormente se  
habilidades de las variabble bidimensional containa y si Z = H₁(X, Y) es una  
Si (X, Y) es una variabble difenentiar ave sóp e algo más complicado. Para  
dimensionally el problema de encontrar ave sóp e algo más complicado. Para  
dimensionally el problema necesitamos un teorema que varnos a indicar y discutir  
a continuacion. Astoes de hacerio, bosquejemos brevemente la idea bisica.

Para encentrar la big de Z = H₁(X, Y) es a mensudo más simple introducir  
una segunda variable aleatoria, digamos W = H₂(X, Y), y obtener primero la fdp  
conjunta de Z y W, digamos g(z), al integrar simplemente k[x, w) como respecto a w.  
Bito es.

g(x) = ∫ · * k[x, w) dw.

Los problemas que quedan son (l) cómo encontrar la sóp conjunta de Z y W, y (2) cieno elegir la variabble aleatoria spropiada W = H₂(X, Y). Para resolver  
el sitimo problema, simplemente, digamos que generalmente hacemos la elección  
más sencina posible de W. En el contexto presente, W desempeta sólo un papel  
intermediancio, y estimente en sons sitorersa en si misma. Con el fin de encostrar  
la fdp conjunta de Z y W secceliamos el teocema 6.3.

Teoremas 6.3. Sepongamos que (X, Y) es una variable aleatoria bidimensional continua con fdp conjunta é, Soa Z = H₁(x, y), y = H₂(x, y), y y = H₂(x, y), y = H₂(x,
```

```
zona_oscura = np.full(input_pic.shape,np.nan)
limite = np.nanmean(input_pic)
for (coor_y,coor_x), value in np.ndenumerate(input_pic):
    aux = mask[coor_y:coor_y+1,coor_x:coor_x+1]
    if aux<limite: #Good Pixel
        zona_oscura[coor_y,coor_x] = input_pic[coor_y,coor_x]
    elif aux>limite: #Bad Pixel
        None
imprimir_pic(zona_oscura,"zona_oscura")
```

zona oscura

```
zona_clara = np.full(input_pic.shape,np.nan)
limite = np.nanmean(input_pic)

for (coor_y,coor_x), value in np.ndenumerate(input_pic):
    aux = mask[coor_y:coor_y+1,coor_x:coor_x+1]
    if aux>limite: #Good Pixel
        zona_clara[coor_y,coor_x] = input_pic[coor_y,coor_x]
    elif aux<=limite: #Bad Pixel
        None

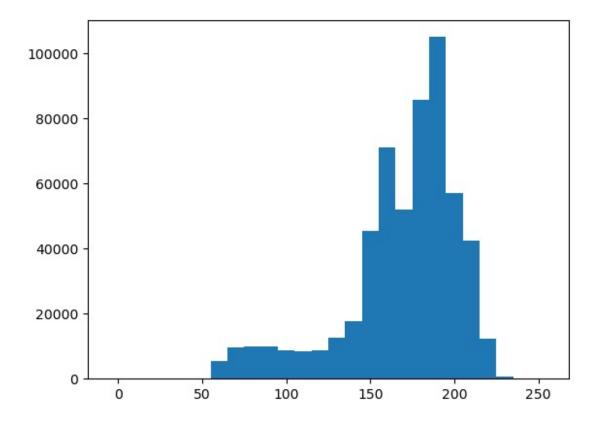
imprimir_pic(zona_clara,"zona_clara")</pre>
```

zona_clara

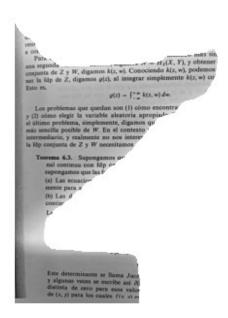
```
Funcionar de una variable alestreria 109

Y=0\circ X=3, Y=0\circ X=4, Y=0\circ X=5, Y=0. Por lo tanto P(U=0)=0.28. El resto de las probabilidades asociadas con U se pueden obtener de una manera El resto de las probabilidades asociadas con U se pueden obtener de una manera surejante. Por tanto, la distribución de probabilidades de U se puede resumir surejante. Por tanto, la distribución de probabilidades de las variables aleatorias V y W como se definió anteriormente se babilidades de las variable aleatorias continua y si Z=H_1(X,Y) es una Si (X,Y) es una variable aleatoria continua (unifonción continua de (X,Y), entonces Z será una variable aleatoria continua (unifonción continua) y el problema de necontrar su fóp es algo más complicado. Para un este problema necesiamos un teorema que varnos a indicar y discutir innación. Antes de hacerto, losquejemos hervemente la lea básica. Per intercoducir y sincurir la fáp de Z=H_1(X,Y) es Z=H_1(X,Y). Para resolver mentrar la fáp de Z=H_1(X,Y) es Z=H_1(X,Y) y Z=H_2(X,Y), y elimitado de Z=H_1(X,Y) es una variable aleatoria bidimensio-duota Z=H_1(X,Y) y Z=H_1(X,Y) y
```

```
\#a,b = np.histogram(zona clara,bins=np.arange(0,270,10))
\#plt.bar(b[0:26],a[0:26],width=10)
\# moda\ zona\ clara = b[np.argmax(a)+1]
#a,b = np.histogram(zona oscura,bins=np.arange(0,270,10))
#plt.bar( b[0:26],a[0:26],width=10)
\#moda\ zona\ oscura = b[np.argmax(a)+1]
osc = np.copy(zona oscura) + ( -moda zona oscura + moda zona clara)
#osc = zona oscura
lessThen0 = osc<0</pre>
moreThen255 = osc>255
osc[lessThen0] = 0
osc[moreThen255] = 255
#imprimir pic(osc, "osc")
a,b = np.histogram(osc,bins=np.arange(0,270,10))
plt.bar(b[0:26],a[0:26],width=10)
\#moda\ zona\ oscura = b[np.argmax(a)+1]
imprimir pic(osc, "Zona oscura con histograma modificado")
imprimir pic(zona clara, "zona clara")
```



Zona oscura con histograma modificado



zona_clara

```
Funcionar de una variable alessararia (10)

y = 0 \circ X = 3, Y = 0 \circ X = 4, Y = 0 \circ X = 5, Y = 0. Por lo tambo P(U = 0) = 0.28. El resuo de las probabilidades asociadas con U se pueden obtener de una manora general succiona. La distribución de prohabilidades de U se puede resumir succiona (1, 2, 3, P(U = u): 0,28, 0,30; 0,25; 0,17. La distribución de prohabilidades de las variables aleatorias verena estabilidades de las variables aleatorias como se definió anteriormente se pueden obsener de una manora semejante. (Ver problema 6.9). Si (X, Y) es una variable bidimensional contanta y si Z = H_1(X, Y) es una sincolo continua de (X, Y), entonces Z será una variable aleatoria continua (unitario continua de exceptamente y si Z = H_1(X, Y)) es problema de encontrar su sópe sa algo más complicado. Para sensonally el problema de encontrar su sópe sa algo más complicado. Para sensonally el problema de encontrar su sópe sa algo más complicado. Para sensonally el problema de encontrar su sópe sa algo más complicado. Para sensonally el problema de encontrar su sópe sa algo más complicado. Para sensonally el problema de encontrar su sópe sa algo más complicado. Para sensonally el problema de encontrar su sópe sa algo más condiciones el problema contrar la sóp de Z = H_1(X, Y) es Z = H_2(X, Y). Para resolver para la sóp de Z = H_1(X, Y) es Z = H_2(X, Y) y Z = H_2
```

```
z1 = np.nan_to_num(zona_clara,0)
z2 = np.nan_to_num(osc,0)

suma_parcial = (z1 + z2)
imprimir_pic(suma_parcial, "combianadas")
```

combianadas

```
Y = 0 o X = 3, Y = 0 o X = 4, Y = 0 o X = 5, Y = 0. Por lo tanto P(U = 0) = 0.28. Y = 0.0 X = 3, Y = 0.0 X = 5, Y = 0. Por lo tanto P(U = 0) = 0.28. Bit sto de las probabilidades asociadas con U se pueden obtener de una manera Bit sto de las probabilidades asociadas con U se pueden obtener de una manera serigiante. Por tanto, la distribución de probabilidades de U se puede resumir stroigante. Por tanto, la distribución de probabilidades de las variables adeatorias V y W como se definió anteriormente se habilidades de las variables adeatorias V y W como se definió anteriormente se habilidades de las variables adeatorias vivil y Bit y
```

```
resta_parcial = np.copy(suma_parcial)
mask_gris = ndimage.median_filter(suma_parcial,
size=15,mode='reflect')

resta_parcial = ((suma_parcial - mask_gris)+255)/(2)
lessThen0 = resta_parcial<0
moreThen255 = resta_parcial>255
resta_parcial[lessThen0] = 0
resta_parcial[moreThen255] = 255

imprimir_pic(resta_parcial, "resta")
p_indicado = contraste_optimo_por_MSE(255 - resta_parcial, 255 - input_pic)
final_metodo_A = funcion_de_contraste(resta_parcial, p_indicado)
imprimir_pic(final_metodo_A, "final_metodo_A")
```

y=0 o X=3, Y=0 o X=4, Y=0 o X=5, Y=0. Por lo tanto P(U=0)=0.23. If sees de las probabilidades associadas con U se pundon obtener de una minera surquinte. Por tanto, is distribución de probabilidades de U se psecés examinariamen en (0,1,3,3); P(U=0); 0.28; 0.30; 0.25; 0.17. La distribución de procaso sigues en (0,1,3,3); P(U=0); 0.28; 0.30; 0.25; 0.17. La distribución de procaso el goberno de las variables destadorias V y V como se definió anteriormente se helicidad en de la variable biblicancian continua y si $Z=H_1(X,Y)$ es una facilita continua de (X,Y); entonces Z será una variable alentoria continua (unidosessimal) y el problema de encontrar se S objecto y exposa o indicar y discoutir a continuación. Asses de hacerlo, benquejemos brevenente la sica hairia.

Para recentar la U de $Z=H_1(X,Y)$ es a menudo más simple introdución una segunda variable abuntos, digitanos $V=H_2(X,Y)$ obtener primero la U0 conjunta de Z y V, degrama X1, w). Conocciondo X1, w), podemos entonces obtener la U1 o U2, U3, U4 integras simplemente V4, V5, V6 con respecto a V5.

Les problèmes que queden son (I) como encontrar la sip conjunta de $Z \neq W_s$ (D) como elegir la variable alrateria apropinda $W = H_2(X,T)$. Para resolver el situino problème, simplemente, d'ègnasse que peneralmente bacemos la elección más senoña posible de W. En el contrato presente, W desempeda sólo un papel jutermeliaria, y ralmente so nos interesa en si misma. Con el fin de encontrar la filo conjunta de $Z \neq W$ accesitamos el teorema 63.

nma 6.3. Seponyamos que (X,Y) es una variable aleatoria bidimensional continua con fáp conjunta f. Sea $Z=H_0(X,Y)$ y $W=H_2(X,Y)$, y uponyamos que has funciones H_1 y H_2 satisfacen las condiciones signicules: (i) Las scurdores $z=H_1(x,y)$ y $w=H_2(x,y)$ so pueden resolver unicamente para x y y on function de x y w, digamen $x=G_1(x,w)$ y $y=G_2(x,w)$. (b) Las delivadas pareiales $\partial x/\partial x$, $\partial x/\partial w$, $\partial y/\partial x$, y $\partial y/\partial w$ existen y son continues.

$$f(z, w) = \begin{vmatrix} \frac{\partial x}{\partial z} & \frac{\partial x}{\partial w} \\ \frac{\partial y}{\partial z} & \frac{\partial y}{\partial w} \end{vmatrix}$$

Esta Jocobiaso de la transformación $(x, y) \rightarrow (t, w)$ acribo sal $d(t, y)(\theta(t, w))$. Observemos que k(x, w) será a esces valores de (x, w) que corresponden a valores les f(x, w) esta destina de $x \rightarrow \infty$.

final_metodo A

Funciones de una variable alcatoria 109

Festivas de est variable interest. 109 $Y = 0 \circ X = 3$, $Y = 0 \circ X = 4$, $Y = 0 \circ X = 5$, Y = 0. Por lo tamto P(U = 0) = 0, 28. El esto de las probabilidades asociadas con U se pueden obtener de una mansera semigiante. Por tanto, la distribución de probabilidades de U se puede resumirante en (0, 1, 2),

$$g(z)=\int_{-\infty}^{\infty}k(z,w)\,dw.$$

Los problemas que quedan son (I) cômo encontrar la fide conjunta de Z y W, y (2) cêmo elegir la variable aleatoria apropiada $W=H_2(X,Y)$. Para resolver el áltimo problema, simplemente, digiamos que generalmente hacemons la elección raía sencifia ponible de W. En el contexto presente, W desempeña sólo un papel intermediani, y realmente no nos internea en si misma. Con el fin de encontrar la fip conjunta de Z y W necesitamos el teorema 6.3.

Terretta 6.3. Supongamos que (X,Y) es una variable aleatoria bidimensional costinua con fdp conjunta f. Sea $Z=H_1(X,Y)$ y $W=H_2(X,Y)$, y supongamos que las funciones H_1 y H_2 satisfacen las condiciones siguientes: (a) Las ecuaciones $x=H_1(x,y)$ $y=H_2(x,y)$ se pueden resolver ûnicamente para x y y en función de z y w, digames $x=G_1(x,w)$ y $y=G_2(x,w)$. (b) Las derivadas parciales $\partial x/\partial x$, $\partial x/\partial w$, $\partial y/\partial x$, y $\partial y/\partial w$ existen y son continua.

Luego la fdp conjunta de (Z,W), digamos h(z,w), está dada por la especsión siguients: $h(z,w)=f[G_1(z,w),\ G_2(z,w)]\ |J(z,w)|$, en donde J(z,w) es el siguiente determinante 2×2 :

$$J(z, w) = \begin{vmatrix} \frac{\partial x}{\partial z} & \frac{\partial x}{\partial w} \\ \frac{\partial y}{\partial z} & \frac{\partial y}{\partial z} \end{vmatrix}$$

Este determinante se llama Jacobiano de la transformación $(x,y) \rightarrow (x,w)$ y algunas veces se escribe asi $\hat{c}(x,y)/\hat{a}(\xi,w)$. Observemos que k(x,w) será dostiata de cero para esto valores de (x,w) que corresponden a valores de (x,y) para los cuales $\ell(x,w)$ es distinto de $a_{\ell(x,y)}$ para los cuales $\ell(x,w)$ es distinto de $a_{\ell(x,y)}$

plt.figure(figsize=(15,12)) plt.subplot(1,4, 1)

```
plt.imshow(input pic, cmap = 'gray')
plt.title('original')
plt.axis('off')
plt.subplot(1,4, 2)
plt.imshow(final metodo BASE, cmap = 'gray')
plt.axis('off')
plt.title('metodo BASE')
plt.subplot(1,4, 3)
plt.imshow(final metodo A, cmap = 'gray')
plt.axis('off')
plt.title('metodo A')
plt.subplot(1,4, 4)
plt.imshow(threshold image, cmap = 'gray')
plt.axis('off')
plt title('metodo B')
plt.show()
       original
                       metodo BASE
                                          metodo A
                                                            metodo B
Guardamos el mejor resultado como archivo jpeg
type(threshold image)
PIL.Image.Image
nombre = imagen entrada + " PROCESADA.jpeg"
```

threshold image.save(nombre)