Importer la PIL package:

**from PIL import Image, ImageFilter**

lecture l'image:

**im = Image.open( 'image.jpg' )**

Afficher l’image

**im.show()**

Filtrer l’iamge

**im\_sharp = im.filter( ImageFilter.SHARPEN )**

Registrer l’image

**im\_sharp.save( 'image\_sharpened.jpg', 'JPEG' )**

Diviser l’image ébranlé RGB

**r,g,b = im\_sharp.split()**

Fusionner l’image

**im = Image.merge("RGB", (b, g, r))**

Redimensionner l’image

**Nouvelle\_image = im.resize(2\* np.array(im.size))**  # resize the matrix

Transformer l’image vers gris

**girs = img.convert(‘L’)**

La mode est optionnelle:

1 (1-bit pixels, black and white, stored with one pixel per byte)

L (8-bit pixels, black and white)

P (8-bit pixels, mapped to any other mode using a colour palette)

RGB (3x8-bit pixels, true colour)

RGBA (4x8-bit pixels, true colour with transparency mask)

CMYK (4x8-bit pixels, colour separation)

YCbCr (3x8-bit pixels, colour video format)

I (32-bit signed integer pixels)

Otsu:

|  |
| --- |
| **from** PIL **import** Image **import** matplotlib.pyplot **as** plt **import** matplotlib.cm **as** cm  **def** histogram(im):  pix =im.load()  width, height = im.size  hist = [0]\*256  **for** y **in** range(height):  **for** x **in** range(width):  gray\_level= pix[x, y]  hist[gray\_level] = hist[gray\_level]+1  **return** hist  **def** otsu\_thrd(im):  hist = histogram(im)  sum\_all = 0  **for** t **in** range(256):  sum\_all += t \* hist[t]  sum\_back, w\_back, w\_for, var\_max, threshold = 0, 0, 0, 0, 0  width, height = im.size  total = height\*width   **for** t **in** range(256):w\_back += hist[t]  **if** (w\_back == 0): **continue** w\_fore = total - w\_back  **if** (w\_fore == 0) : **break** *# calculate classes means* sum\_back += t \* hist[t]  mean\_back = sum\_back / w\_back  mean\_fore = (sum\_all - sum\_back) / w\_fore  *# Calculate Between Class Variance* var\_between = w\_back \* w\_fore \* (mean\_back - mean\_fore)\*\*2  **if** (var\_between > var\_max):  var\_max = var\_between  threshold = t  **return** threshold  **def** segment(im, thrd = 128):  width, height = im.size  mat = im.load()  out = Image.new(**'1'**,(width, height))  out\_pix = out.load()  **for** x **in** range(width): *# go over the image columns* **for** y **in** range(height): *# go over the image rows* **if** mat[x, y] >= thrd: *# compare to threshold* out\_pix[x, y] = 255  **else**:  out\_pix[x, y] = 0  **return** out  im=Image.open(**'10.JPG'**).convert(**'L'**) th=otsu\_thrd(im) im\_out=segment(im,th) plt.imshow(im\_out, cmap=cm.gray) plt.axis(**"off"**) plt.show() |

diviser les paragraphes en lignes: