Las transformaciones morfológicas son algunas operaciones simples basadas en la forma de la imagen. Normalmente se realiza en imágenes binarias. Necesita dos entradas, una es nuestra imagen original, la segunda se llama elemento estructurante o núcleo que decide la naturaleza de la operación. Dos operadores morfológicos básicos son la Erosión y la Dilatación. Entonces también entran en juego sus variantes como Apertura, Cierre, Degradado, etc. Los veremos uno por uno con la ayuda de la siguiente imagen:

import cv2 as cv  
import numpy as np  
img = cv.imread('resources/j.png', cv.IMREAD\_GRAYSCALE)  
assert img is not None, "file could not be read, check with os.path.exists()"  
kernel = np.ones((5,5),np.uint8)  
erosion = cv.erode(img,kernel,iterations = 1)

1. Dilatación

Es justo lo contrario de la erosión. Aquí, un elemento de píxel es '1' si al menos un píxel bajo el núcleo es '1'. Por lo tanto, aumenta la región blanca en la imagen o aumenta el tamaño del objeto en primer plano. Normalmente, en casos como la eliminación de ruido, a la erosión le sigue la dilatación. Porque la erosión elimina los ruidos blancos, pero también encoge nuestro objeto. Entonces lo dilatamos. Como el ruido desaparece, no volverán, pero el área de nuestro objeto aumenta. También es útil para unir partes rotas de un objeto.

dilation = cv.dilate(img,kernel,iterations = 1)

1. Apertura

La apertura es sólo otro nombre de la erosión seguida de dilatación . Es útil para eliminar el ruido, como explicamos anteriormente. Aquí usamos la función cv.morphologyEx()

opening = cv.morphologyEx(img, cv.MORPH\_OPEN, kernel)

1. Cierre

El cierre es lo contrario de la apertura, la dilatación seguida de la erosión . Es útil para cerrar pequeños agujeros dentro de los objetos en primer plano o pequeños puntos negros en el objeto.

closing = cv.morphologyEx(img, cv.MORPH\_CLOSE, kernel)

1. gradiente morfológico

Es la diferencia entre dilatación y erosión de una imagen.

El resultado se parecerá al contorno del objeto.

gradient = cv.morphologyEx(img, cv.MORPH\_GRADIENT, kernel)

1. Sombrero de copa

Es la diferencia entre imagen de entrada y apertura de la imagen. El siguiente ejemplo se realiza para un kernel 9x9.

tophat = cv.morphologyEx(img, cv.MORPH\_TOPHAT, kernel)

1. Sombrero negro

Es la diferencia entre el cierre de la imagen de entrada y la imagen de entrada.

blackhat = cv.morphologyEx(img, cv.MORPH\_BLACKHAT, kernel)

Elemento Estructurante Creamos manualmente elementos estructurantes en los ejemplos anteriores con la ayuda de Numpy. Es de forma rectangular. Pero en algunos casos, es posible que necesites granos de forma elíptica o circular. Entonces, para este propósito, OpenCV tiene una función, cv.getStructuringElement() . Simplemente pasas la forma y el tamaño del grano y obtienes el grano deseado.

# Rectangular Kernel  
cv.getStructuringElement(cv.MORPH\_RECT,(5,5))  
# Elliptical Kernel  
cv.getStructuringElement(cv.MORPH\_ELLIPSE,(5,5))  
# Cross-shaped Kernel  
cv.getStructuringElement(cv.MORPH\_CROSS,(5,5))

array([[0, 0, 1, 0, 0],  
 [0, 0, 1, 0, 0],  
 [1, 1, 1, 1, 1],  
 [0, 0, 1, 0, 0],  
 [0, 0, 1, 0, 0]], dtype=uint8)