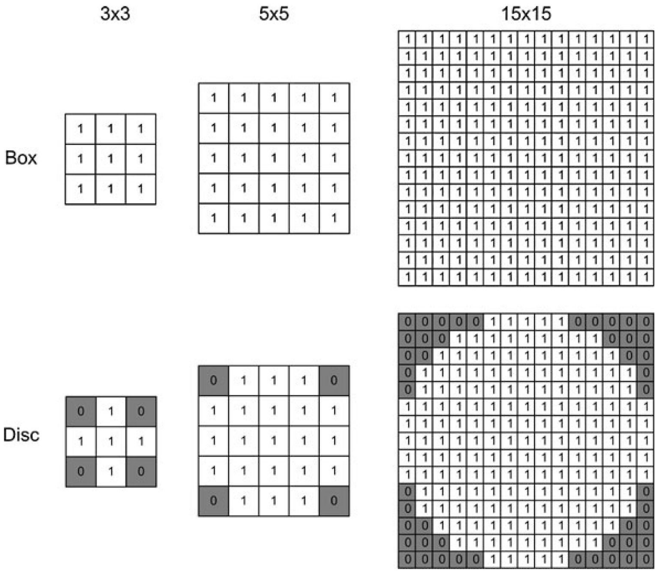
# Lab 3
# Pengolahan Citra
# Morphological Image Processing
# Senin, 12 Oktober 2020

## A. Structuring Element

Morphological technique probe an image with a shape called structuring element (or kernel). This  structuring element is positioned at all possible locations in the image and to conclude how this shape fits or misses in the image. The structuring element can be disk-shaped, rectangular, diamond-shaped, or you can define your own structuring element shape and size in a binary matrix.

Here’s an example of structuring element shapes and sizes.



| | | |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 1 | 1 |
| 1 | 1 | 1 |

| | | | | |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| | | |
|---|---|---|
| 0 | 1 | 0 |
| 1 | 1 | 1 |
| 0 | 1 | 0 |

| | | | | |
|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 |

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |

Source: https://medium.com/@himnickson/morphological-operations-in-image-processing-cb8045b98fcc

## B. Basic Operators

### 1. Dilation

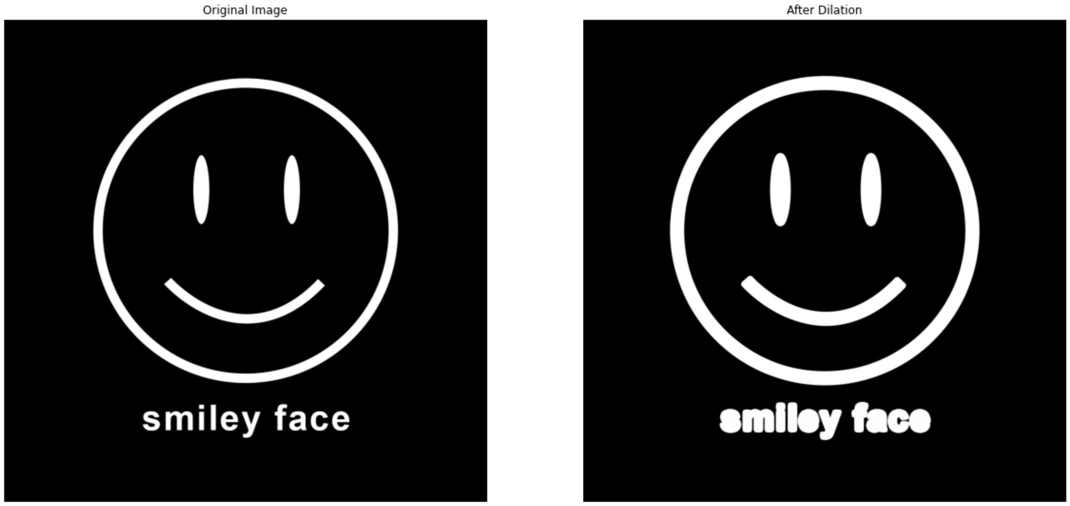
Dilation (usually represented by ⊕) is a process of adding pixels to the boundaries of image. After dilation operation, the image will appear thicker than before. Dilation takes two inputs, the image that will be dilated and the structuring element (also known as kernel).

```
from skimage import io, color
from skimage.morphology import disk, dilation
import matplotlib.pyplot as plt

# Import and change image to gray
im = color.rgb2gray(io.imread('smiley.png'))

# You can change the structuring element (se)
# Read skimage.morphology documentation for more
se = disk(5)
dilated = dilation(im, se)

# Show image
plt.figure(figsize=(20,10))
plt.subplot(1,2,1)
plt.title('Original Image')
plt.imshow(im, cmap='gray')
plt.axis('off')
plt.subplot(1,2,2)
plt.title('After Dilation')
plt.imshow(dilated, cmap='gray')
plt.axis('off')
plt.show()
```

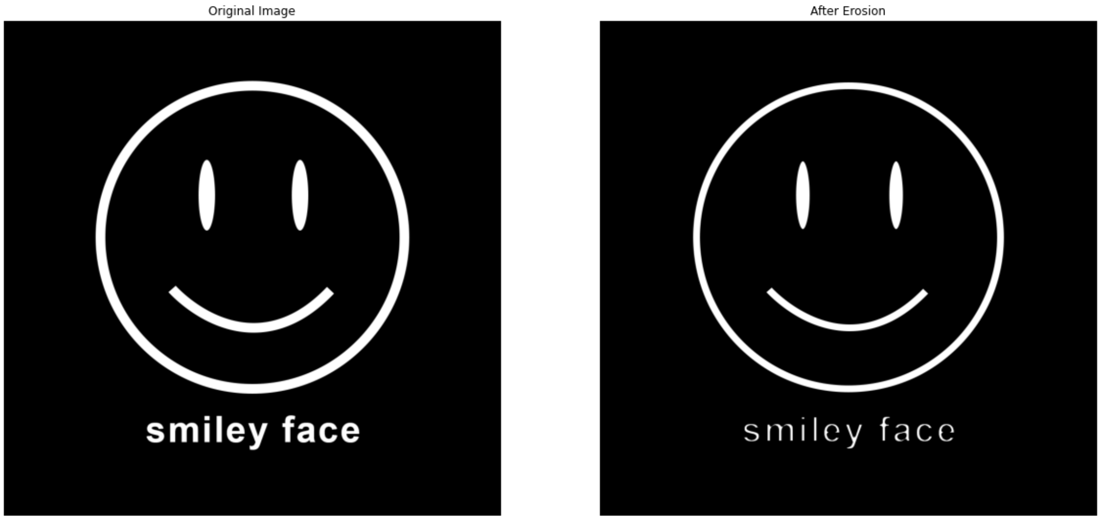


## 2. Erosion

Erosion (usually represented by ⊖) is the opposite of dilation. It removes pixels on object boundaries. The number of pixels removed from the objects depends on the size and shape of the structuring element used. This operation will result thinner image.

```
from skimage.morphology import disk, erosion

# You can change the structuring element(se)
# Read skimage.morphology documentation for more
se = disk(2)
eroded = erosion(im, se)
```



Original Image

After Erosion

To get a better visualization of erosion and dilation process, you can see the illustration here.

### 3. Opening

Opening is an operator derived from the dilatation and erosion operation. To be ecxact, opening is process of erosion followed by a dilation using the same structuring element. Morphological opening is useful for removing small objects from an image while preserving the shape and size of larger objects in the image.

Can be written in the equation below,

$$A \circ B = (A \ominus B) \oplus B$$

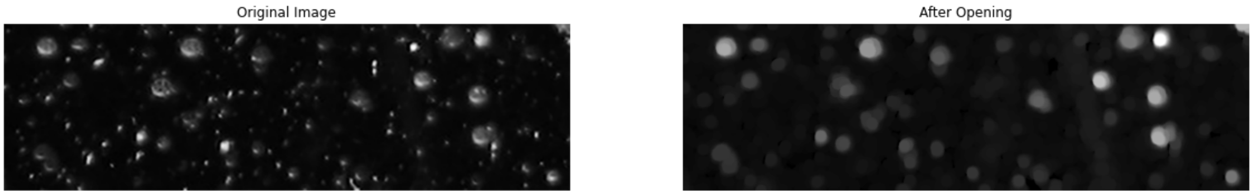
where A is the original image, B is the structuring element, $\oplus$ is dilation, and $\ominus$ is erosion.

```
from skimage import io, color
from skimage.util import img_as_ubyte
from skimage.morphology import disk, opening
```

```
import matplotlib.pyplot as plt

# Change image to gray
im = color.rgb2gray(io.imread('snowflakes.jpg'))

# You can change the structuring element
# Read skimage.morphology documentation for more
se = disk(10)
open = opening(im, se)
```



Original Image

After Opening

source: https://in.mathworks.com/help/images/granulometry-of-snowflakes.html

4. **Closing**

   Closing is the opposite of opening. It’s a process of dilation followed by erosion using the same structuring element for both operations. Closing is useful for filling small holes from an image while preserving the shape and size of the objects in the image.

   Can be written in the equation below,

$$A \bullet B = (A \oplus B) \ominus B$$

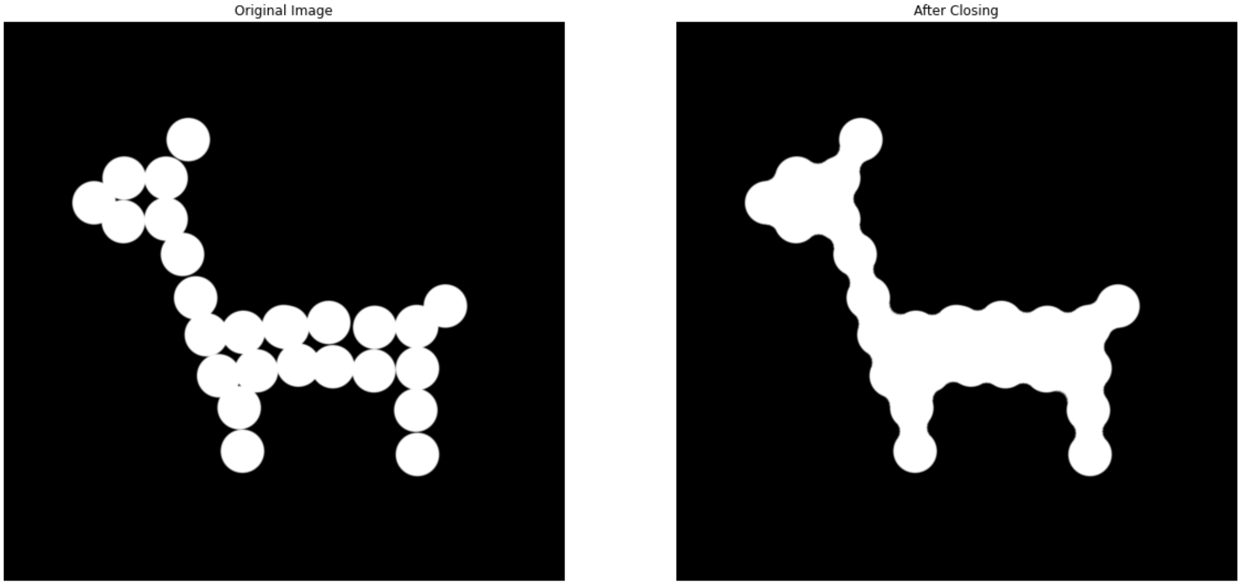
   where A is the original image, B is the structuring element, $\oplus$ is dilation, and $\ominus$ is erosion.

```
from skimage import io, color
from skimage.morphology import disk, closing

# Change image to gray
im = color.rgb2gray(io.imread('girrafe.png'))

# You can change the structuring element (se)
# Read skimage.morphology documentation for more

se = disk(20)
closed = closing(im, se)
```



## C. Other Operators

### 1. Top-Hat Filter

Usually refers to White Top-Hat Filter. Given an image, top-hat transform will extract small elements and details which are brighter than its surroundings. It is useful in observing minor details of the inputs when they are present as light pixels on a dark background. Top-Hat is obtained from the difference between the input image and its opening by some structuring element (see equation below).

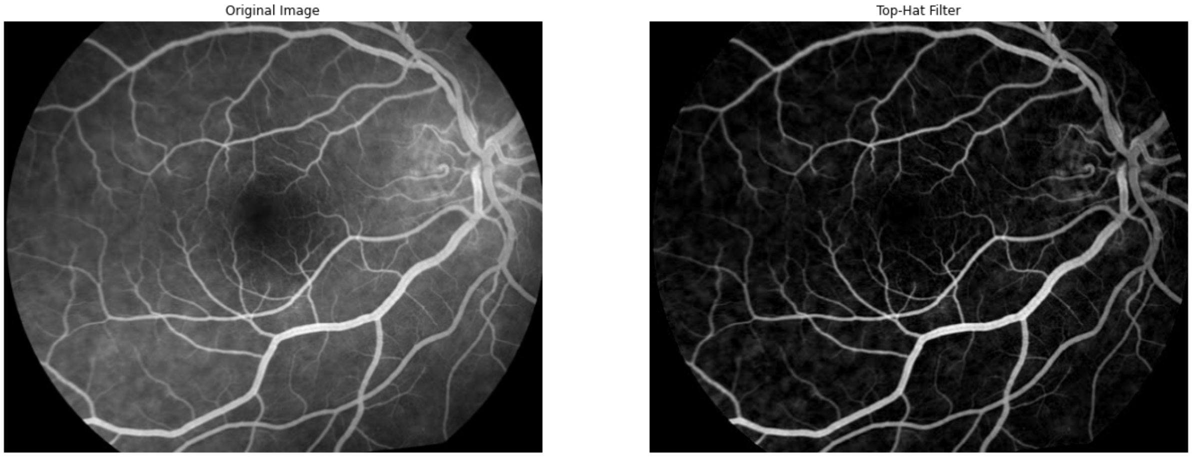
$$T_H(f) = f - f \circ b$$

```
from skimage import io, color
from skimage.util import img_as_ubyte
from skimage.morphology import disk, white_tophat
import matplotlib.pyplot as plt

# Change image to RGB
im = color.rgb2gray(io.imread('retina.jpg'))

# You can change the structure element
# Read skimage.morphology documentation for more
se = disk(20)
tp = white_tophat(im, se)

# Show image
plt.figure(figsize=(20,10))
plt.subplot(1,2,1)
plt.title('Original Image')
plt.imshow(im, cmap='gray')
plt.axis('off')
plt.subplot(1,2,2)
plt.title('Top-Hat Filter')
plt.imshow(tp, cmap='gray')
plt.axis('off')
plt.show()
```



Source: http://www.retinareference.com/anatomy/

### 2. Bottom-Hat Filter

Refers to as Black Top-Hat Filter. Bottom-hat extracts small elements and details which are darker than its surroundings. It is useful in observing minor details of the inputs when they are present as dark pixels on a light background. Bottom-Hat is obtained from the difference between the input image closing and its original one by some structuring element (see equation below).

$$B_H(f) = f \bullet b - f$$

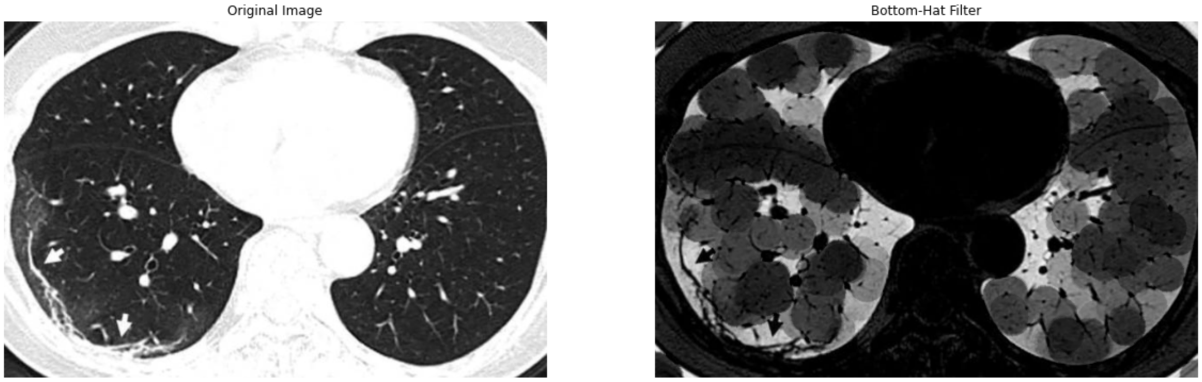
```
from skimage import io, color
from skimage.util import img_as_ubyte
from skimage.morphology import disk, black_tophat
import matplotlib.pyplot as plt

# Change image to RGB
im = color.rgb2gray(io.imread('covid_lung.jpg'))

# You can change the structure element
# Read skimage.morphology documentation for more
se = disk(20)
```

```
bt = black_tophat(im, se)

# Show image
plt.figure(figsize=(20,10))
plt.subplot(1,2,1)
plt.title('Original Image')
plt.imshow(im, cmap='gray')
plt.axis('off')
plt.subplot(1,2,2)
plt.title('Bottom-Hat Filter')
plt.imshow(bt, cmap='gray')
plt.axis('off')
plt.show()
```



Source: https://www.whitsundaytimes.com.au/news/shocking-photo-of-virus-killing-patient/3969985/

**Warning!**

**There are also modules for bottom-hat and top-hat filtering from skimage.filters.rank. The modules are actually deprecated, not recommended to be used.**