

# Shri Ramdeobaba College of Engineering and Management

## Nagpur, 440013

### Department of Computer Science Engineering

FDVIP Lab

**Name :** *Shantanu Mane*

**Roll No. :** *E63*

**Batch :** *CSE-AIML*

**Date :** *5/4/2023*

**AIM -** To study and perform morphological operations on an image.

1. Erosion
2. Dilation
3. Opening
4. Closing

## Importing Dependencies

```
import cv2
import matplotlib.pyplot as plt
import numpy as np
```

## Reading the images

```
image = cv2.imread("../data/mri_2.png", 0)
```

## 1. Erosion

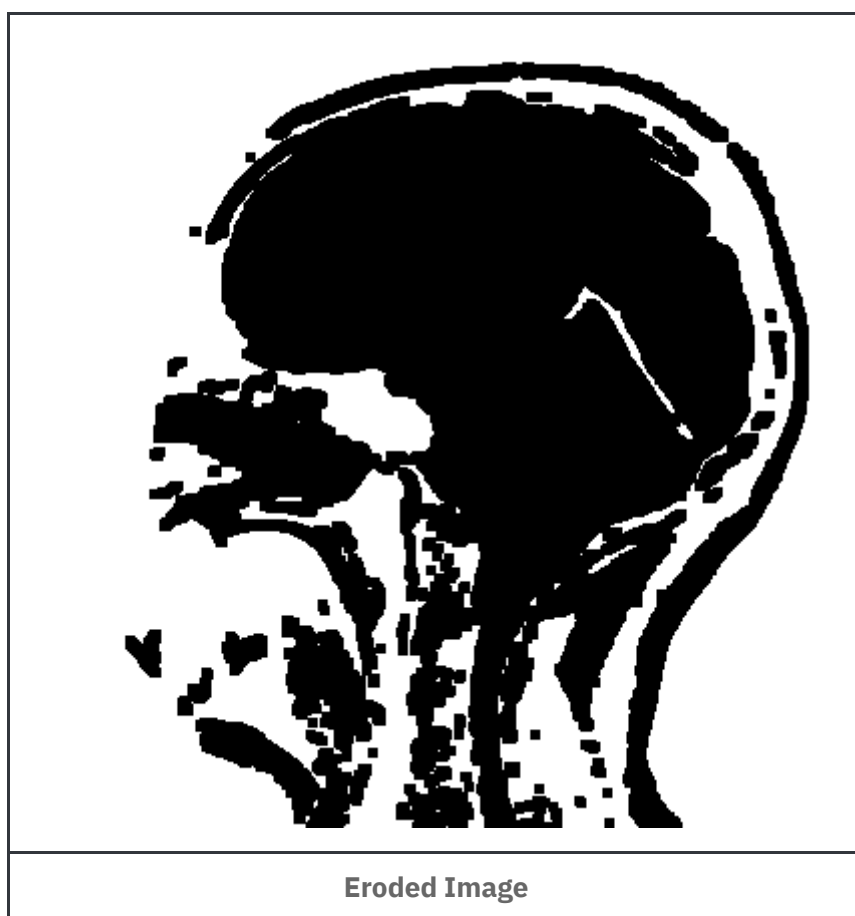
## Creating Kernel

```
kernel = np.ones((5, 5), np.uint8)
```

## Applying Erosion

```
binr = cv2.threshold(img, 0, 255, cv2.THRESH_BINARY + cv2.THRESH_OTSU)[1]  
invert = cv2.bitwise_not(binr)  
erosion = cv2.erode(invert, kernel, iterations=1)
```

## Output



## 2. Dilation

### Creating Kernel

```
kernel = np.ones((5, 5), np.uint8)
```

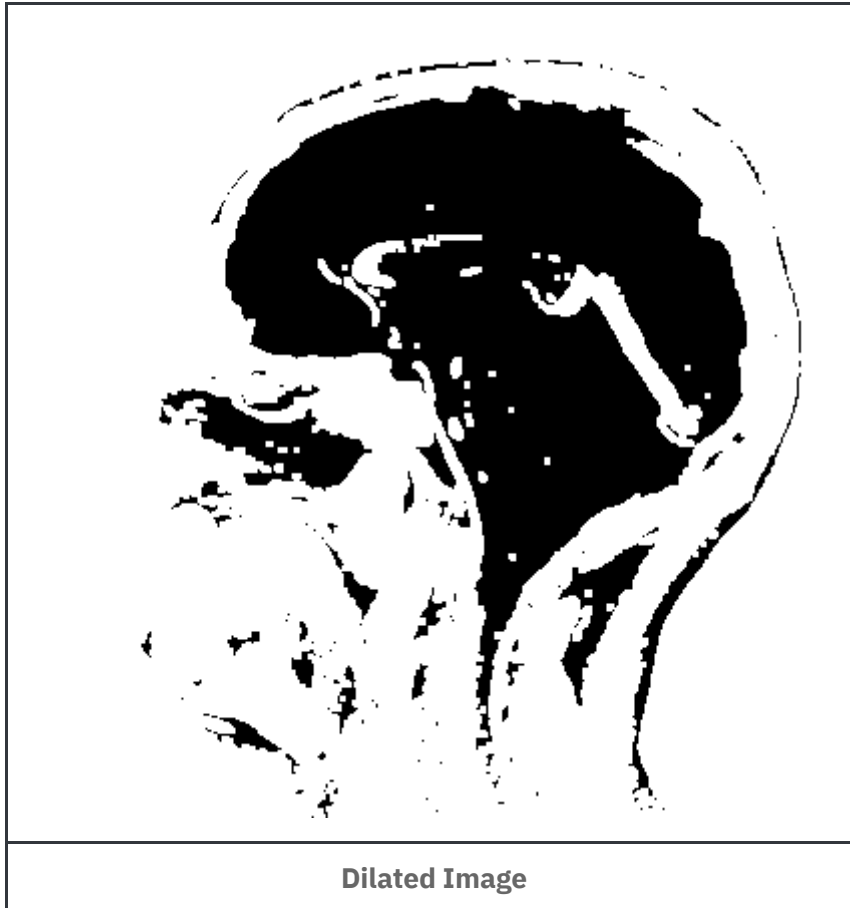
### Applying Dilation

```
binr = cv2.threshold(img, 0, 255, cv2.THRESH_BINARY + cv2.THRESH_OTSU)[1]

invert = cv2.bitwise_not(binr)

dilation = cv2.dilate(invert, kernel, iterations=1)
```

## Output



## 3. Opening

### Creating Kernel

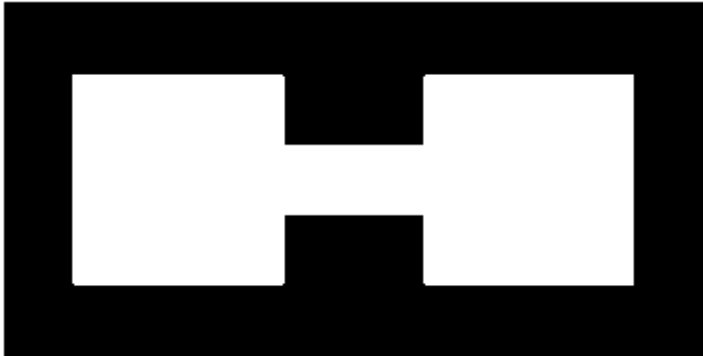
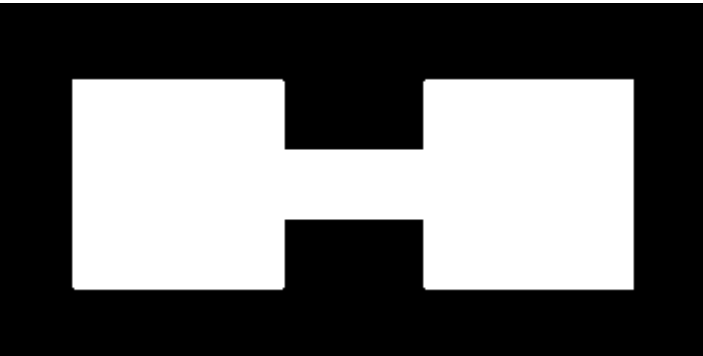


```
kernel = np.ones((5, 5), np.uint8)
```

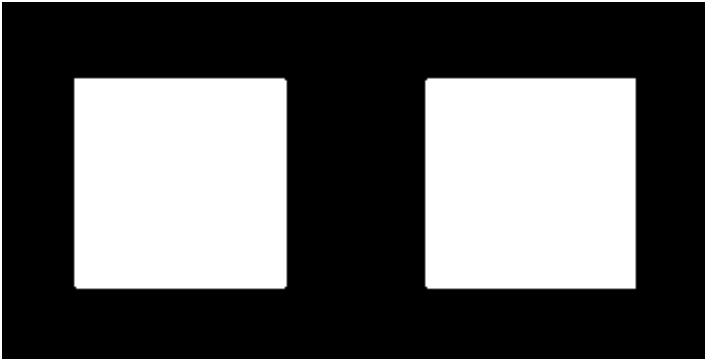

### Applying Opening

```
binr = cv2.threshold(img, 0, 255, cv2.THRESH_BINARY + cv2.THRESH_OTSU)[1]

invert = cv2.bitwise_not(binr)
for i in range(0, 25, 5):
    opening = cv2.morphologyEx(binr, cv2.MORPH_OPEN, kernel, iterations=i)
    cv2.imwrite(f"../data/opening_{i}.png", opening)
```

Output

Iterations	Image
0	 A 28x28 pixel binary image showing a noisy representation of the digit '2'. The background is black, and the digit is white. There are several black pixels scattered within the white area of the digit, particularly in the central horizontal bar and the top and bottom loops.
5	 A 28x28 pixel binary image showing a noisy representation of the digit '2'. The background is black, and the digit is white. There are several black pixels scattered within the white area of the digit, particularly in the central horizontal bar and the top and bottom loops.
10	 A 28x28 pixel binary image showing a clean representation of the digit '2'. The background is black, and the digit is white. The digit is composed of two main vertical strokes connected by a horizontal bar in the middle. The top and bottom loops are also clearly defined.
15	 A 28x28 pixel binary image showing a clean representation of the digit '2'. The background is black, and the digit is white. The digit is composed of two main vertical strokes connected by a horizontal bar in the middle. The top and bottom loops are also clearly defined.

Iterations	Image
20	
25	

## 4. Closing

### Creating Kernel

```
kernel = np.ones((5, 5), np.uint8)
```

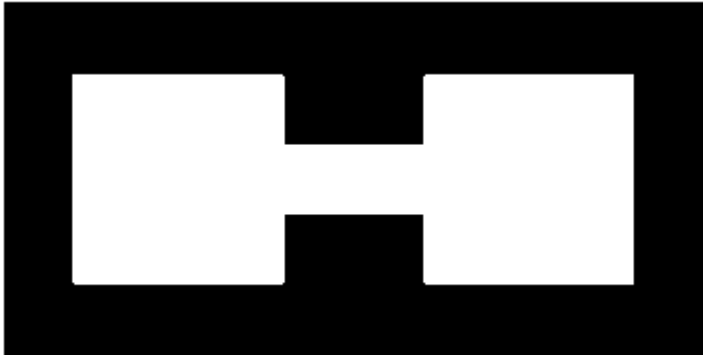
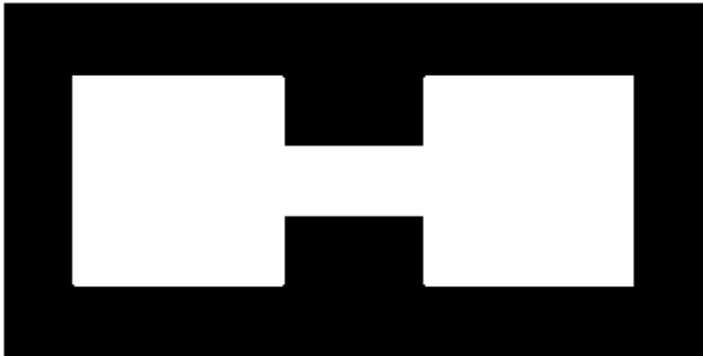



### Applying Closing

```
binr = cv2.threshold(img, 0, 255, cv2.THRESH_BINARY + cv2.THRESH_OTSU)[1]

invert = cv2.bitwise_not(binr)
for i in range(0, 25, 5):
    closing = cv2.morphologyEx(binr, cv2.MORPH_CLOSE, kernel, iterations=i)
    cv2.imwrite(f"../data/closing_{i}.png", closing)
```

### Output

Iterations	Image
------------	-------

Iterations	Image
0	
5	
10	
15	
20	

Iterations	Image
25	