

Q2: Regulated Dilation

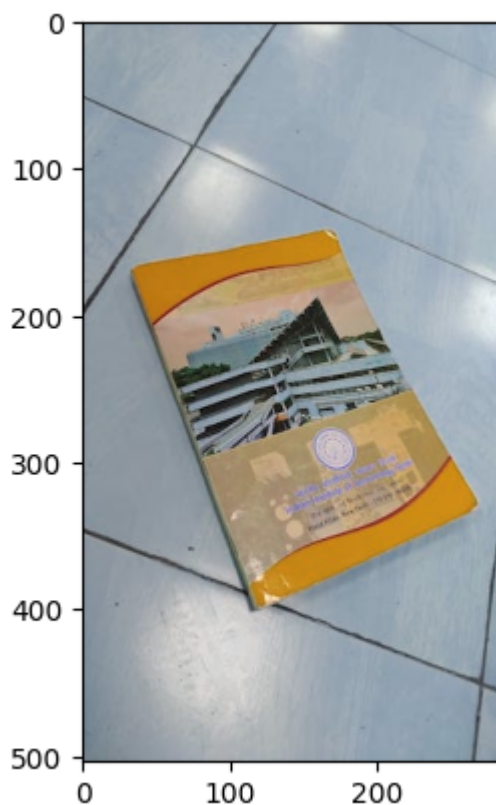
$$A' = (A+B)-B$$

Take an image of a book placed on white background and show the output of this operation, you can report the result with square, rectangle, circle, diamond, plus sign structuring element

```
In [1]: import cv2 as cv
import numpy as np
import matplotlib.pyplot as plt
```

```
In [2]: img = cv.imread("book.jpg")
plt.imshow(img)
```

```
Out[2]: <matplotlib.image.AxesImage at 0x17de11e50>
```



```
In [3]: square = np.array([
    [1,1,1,1,1,1,1],
    [1,1,1,1,1,1,1],
    [1,1,1,1,1,1,1],
    [1,1,1,1,1,1,1],
    [1,1,1,1,1,1,1],
    [1,1,1,1,1,1,1],
    [1,1,1,1,1,1,1],
    [1,1,1,1,1,1,1],
], dtype=np.uint8)

rectangle = np.array([
    [0,0,0,0,0,0,0],
    [0,0,1,1,1,0,0],
```

```

    [0,0,1,1,1,0,0],
    [0,0,1,1,1,0,0],
    [0,0,1,1,1,0,0],
    [0,0,1,1,1,0,0],
    [0,0,0,0,0,0,0],
], dtype=np.uint8)

circle = np.array([
    [0,0,1,1,1,0,0],
    [0,1,1,1,1,1,0],
    [1,1,1,1,1,1,1],
    [1,1,1,1,1,1,1],
    [1,1,1,1,1,1,1],
    [0,1,1,1,1,1,0],
    [0,0,1,1,1,0,0],
], dtype=np.uint8)

diamond = np.array([
    [0,0,0,1,0,0,0],
    [0,0,1,1,1,0,0],
    [0,1,1,1,1,1,0],
    [1,1,1,1,1,1,1],
    [0,1,1,1,1,1,0],
    [0,0,1,1,1,0,0],
    [0,0,0,1,0,0,0],
], dtype=np.uint8)

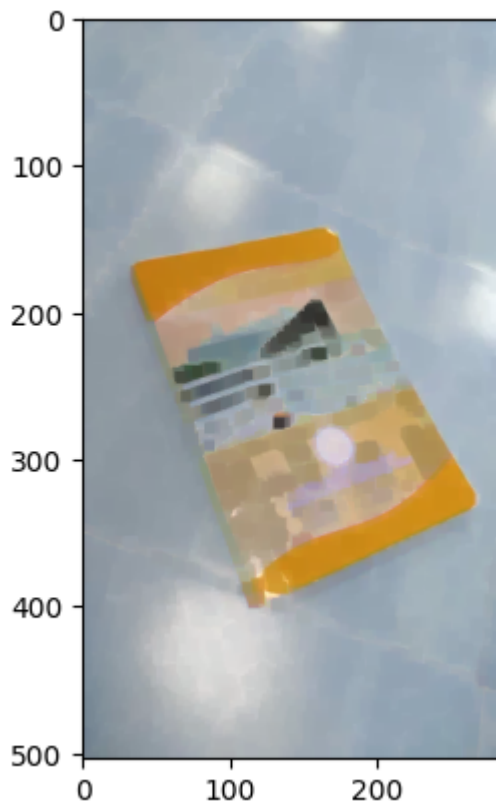
plus = np.array([
    [0,0,0,1,0,0,0],
    [0,0,0,1,0,0,0],
    [0,0,0,1,0,0,0],
    [1,1,1,1,1,1,1],
    [0,0,0,1,0,0,0],
    [0,0,0,1,0,0,0],
    [0,0,0,1,0,0,0],
], dtype=np.uint8)

```

Square

```
In [4]: plt.imshow(cv.erode(cv.dilate(img, square), square))
```

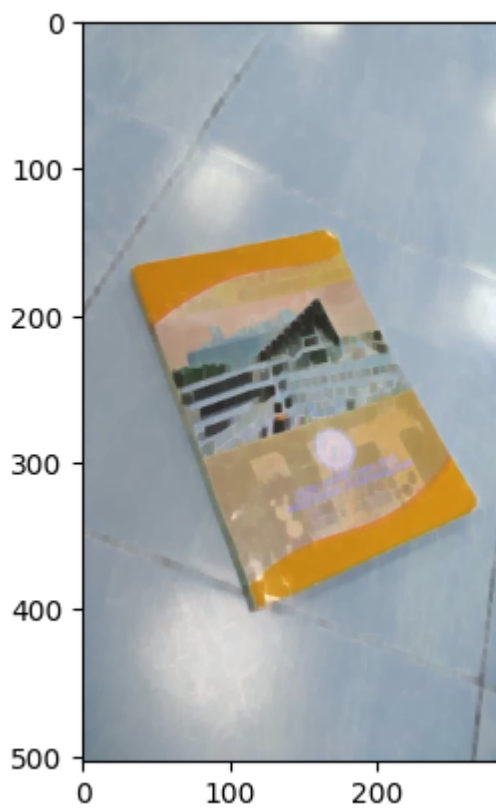
```
Out[4]: <matplotlib.image.AxesImage at 0x17deb3e90>
```



Rectangle

```
In [5]: plt.imshow(cv.erode(cv.dilate(img, rectangle), rectangle))
```

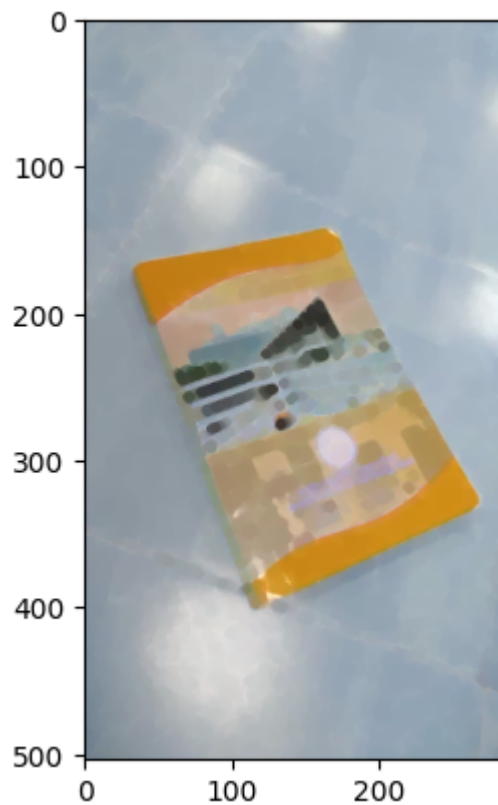
```
Out[5]: <matplotlib.image.AxesImage at 0x17df05f10>
```



Circle

```
In [6]: plt.imshow(cv.erode(cv.dilate(img, circle), circle))
```

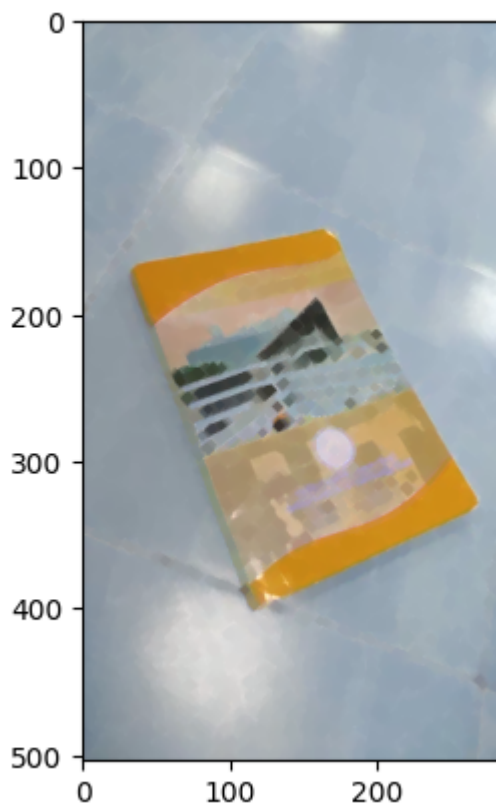
```
Out[6]: <matplotlib.image.AxesImage at 0x17df86a10>
```



Diamond

```
In [7]: plt.imshow(cv.erode(cv.dilate(img, diamond), diamond))
```

```
Out[7]: <matplotlib.image.AxesImage at 0x17dfd3e90>
```



Plus

```
In [8]: plt.imshow(cv.erode(cv.dilate(img, plus), plus))
```

```
Out[8]: <matplotlib.image.AxesImage at 0x17e030ad0>
```

