

## Домашна задача 2

Имплементирајте Compass оператор за детекција на рабови (изберете слика по ваша желба).

(а) Пресметајте и прикажете го резултатот на секој од филтрите.

### Решение:

```
import cv2
import numpy as np

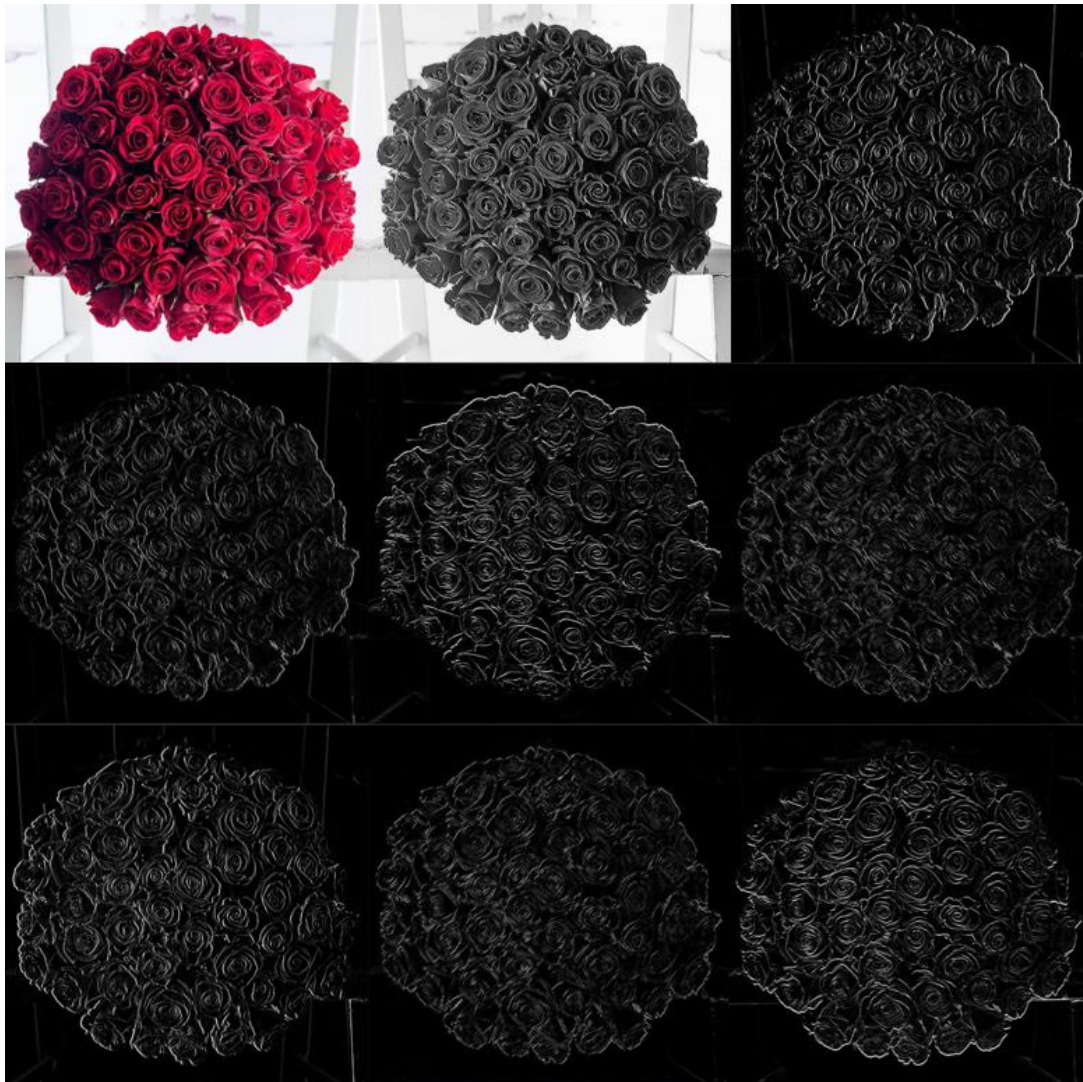
img = cv2.imread('image.jpg')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

n_mask = np.array([[ -1, -1, -1],
                   [ -1,  2, -1],
                   [  1,  1,  1]], dtype=np.float32)
ne_mask = np.array([[ -1, -1,  2],
                   [ -1,  2, -1],
                   [  2, -1, -1]], dtype=np.float32)
e_mask = np.array([[  1, -1, -1],
                   [  1,  2, -1],
                   [  1, -1, -1]], dtype=np.float32)
se_mask = np.array([[  2, -1, -1],
                   [ -1,  2, -1],
                   [ -1, -1,  2]], dtype=np.float32)
s_mask = np.array([[  1,  1,  1],
                   [ -1,  2, -1],
                   [ -1, -1, -1]], dtype=np.float32)
sw_mask = np.array([[ -1,  2, -1],
                   [ -1,  2, -1],
                   [ -1, -1,  2]], dtype=np.float32)
w_mask = np.array([[ -1, -1,  1],
                   [ -1,  2,  1],
                   [ -1, -1,  1]], dtype=np.float32)
nw_mask = np.array([[  2, -1, -1],
                   [  1,  2, -1],
                   [ -1, -1,  1]], dtype=np.float32)

n_edges = cv2.filter2D(gray, -1, n_mask)
ne_edges = cv2.filter2D(gray, -1, ne_mask)
```

```
e_edges = cv2.filter2D(gray, -1, e_mask)
se_edges = cv2.filter2D(gray, -1, se_mask)
s_edges = cv2.filter2D(gray, -1, s_mask)
sw_edges = cv2.filter2D(gray, -1, sw_mask)
w_edges = cv2.filter2D(gray, -1, w_mask)
nw_edges = cv2.filter2D(gray, -1, nw_mask)
```

```
cv2.imshow('N', n_edges)
cv2.imshow('NE', ne_edges)
cv2.imshow('E', e_edges)
cv2.imshow('SE', se_edges)
cv2.imshow('S', s_edges)
cv2.imshow('SW', sw_edges)
cv2.imshow('W', w_edges)
cv2.imshow('NW', nw_edges)
cv2.waitKey(0)
cv2.destroyAllWindows()
```



(b) Пресметајте и прикажете го резултатот добиен со комбинација на сите филтри. Тестирајте со различни вредности за прагот.

**Решение:**

```
import cv2
import numpy as np

img = cv2.imread('image.jpg')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

n_mask = np.array([[ -1, -1, -1],
                   [ -1,  2, -1],
                   [  1,  1,  1]], dtype=np.float32)
ne_mask = np.array([[ -1, -1,  2],
                   [ -1,  2, -1],
                   [  2, -1, -1]], dtype=np.float32)
e_mask = np.array([[  1, -1, -1],
                   [  1,  2, -1],
                   [  1, -1, -1]], dtype=np.float32)
se_mask = np.array([[  2, -1, -1],
                   [ -1,  2, -1],
                   [ -1, -1,  2]], dtype=np.float32)
s_mask = np.array([[  1,  1,  1],
                   [ -1,  2, -1],
                   [ -1, -1, -1]], dtype=np.float32)
sw_mask = np.array([[ -1,  2, -1],
                   [ -1,  2, -1],
                   [ -1, -1,  2]], dtype=np.float32)
w_mask = np.array([[ -1, -1,  1],
                   [ -1,  2,  1],
                   [ -1, -1,  1]], dtype=np.float32)
nw_mask = np.array([[  2, -1, -1],
                   [  1,  2, -1],
                   [ -1, -1,  1]], dtype=np.float32)

n_edges = cv2.filter2D(gray, -1, n_mask)
ne_edges = cv2.filter2D(gray, -1, ne_mask)
e_edges = cv2.filter2D(gray, -1, e_mask)
se_edges = cv2.filter2D(gray, -1, se_mask)
s_edges = cv2.filter2D(gray, -1, s_mask)
sw_edges = cv2.filter2D(gray, -1, sw_mask)
w_edges = cv2.filter2D(gray, -1, w_mask)
nw_edges = cv2.filter2D(gray, -1, nw_mask)
```

```

threshold = 150
combined = np.zeros_like(gray)
combined[(n_edges > threshold) |
        (ne_edges > threshold) |
        (e_edges > threshold) |
        (se_edges > threshold) |
        (s_edges > threshold) |
        (sw_edges > threshold) |
        (w_edges > threshold) |
        (nw_edges > threshold)] = 255

cv2.imshow('Compass Edge Detection', combined)
cv2.waitKey(0)
cv2.destroyAllWindows()

```



**Изработил:**

**Бојан Ристов (211151)**