

پاسخ سوال اول

الف) به هر ردیف و ستون یک صفر اضافه میشود و تصویر حاصل $6*6$ میشود. به صورت زیر:

0	0	0	0	0	0
0	1	2	1	6	0
0	7	1	1	1	0
0	3	1	2	0	0
0	1	4	0	2	0
0	0	0	0	0	0

و در حالت replicate به صورت زیر است:

1	1	2	1	6	6
1	1	2	1	6	6
7	7	1	1	1	1
3	3	1	2	0	0
1	1	4	0	2	2
1	1	4	0	2	2

ادامه سوال در صفحه ی بعد...

ابتدا باید فیلتر را 180 درجه بچرخانیم:

1	2	1
1	0	1
11	3	1

Zero padding:

24	13	13	5
15	22	19	16
23	28	11	11
11	8	11	2

BOARDER_REFLECT:

37	19	23	31
26	22	19	23
34	28	11	14
23	21	17	12

مقایسه این دو حالت:

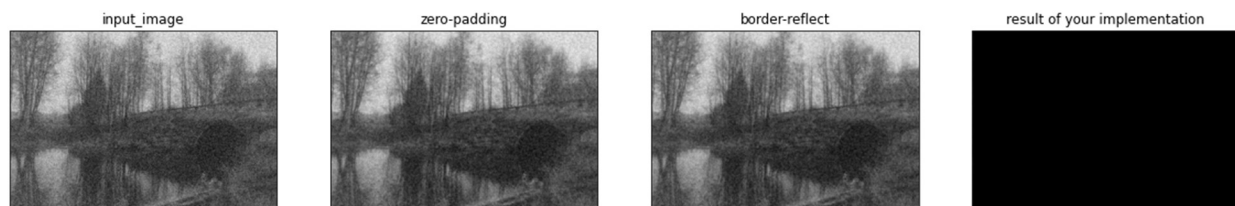
همانطور که در کلاس درس نیز بحث شد، حالت به شکل زیر است:

BORDER_REFLECT	
Python: cv.BORDER_REFLECT	fedcba abcdefgh hgfedcb

[link](#)

نکته ی جالب دیگر، حاشیه تصاویر متفاوت هستند که به دلیل متفاوت بودن نوع padding است و مرکز تصویر دقیقا یکی هستند.

من روی عکس داده شده نیز این مورد را بررسی کردم و تصاویر را از یکدیگر کم کردم نتیجه به شکل زیر بود:



آرایه ی تصویر حاصل نیز بیانگر این مطلب بود:

```
[ [66 42 37 ... 58 48 53]
  [48 0 0 ... 0 0 8]
  [49 0 0 ... 0 0 8]
  ...
  [23 0 0 ... 0 0 3]
  [27 0 0 ... 0 0 4]
  [27 4 3 ... 4 4 7] ]
```

پاسخ سوال دوم

عملگر سوبل در واقع قصد دارد با دو کرنل عملیات مشتق را در دو جهت انجام بدهد. به عنوان مثال اگر آرایه ایی داشته باشیم و بخواهیم مشتق را در هر خانه آرایه حساب کنیم، مشتق یعنی مقدار خانه قبلی را از مقدار خانه بعدی کم کنیم که کرنلش به صورت: $[+1, 0, -1]$ (مشتق در راستای x) چون این حالت قرینه کرنل اولی داده شده در سوال است فکر میکنم ممکن است مربوط به لاپلاسین گیری باشد. این کرنل تصویر را تیزتر میکند بخصوص برای نشان دادن جزئیات عمودی تصویر.

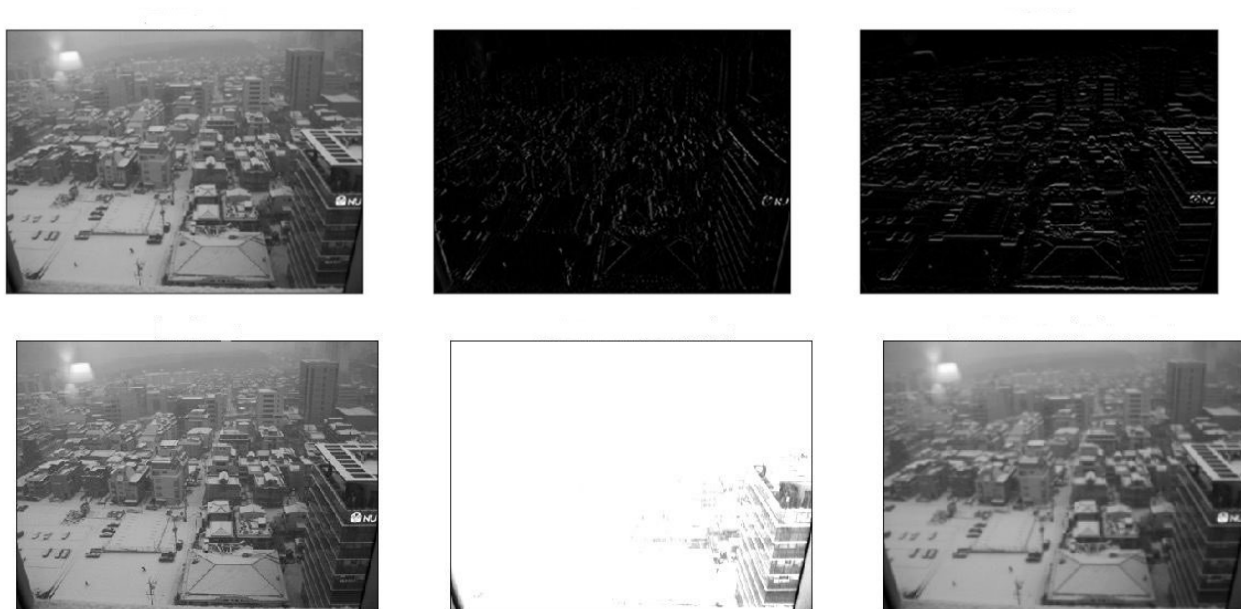
برای اینکار لازم است کرنل بالا را با داده های آرایه کانولوشن کنیم. به این ترتیب عملیات مشتق بر روی آرایه صورت میگیرد. کار دومی که انجام میگیرد در عملگر سوبل رفع نویز هست که میتوان این کار را با میانگین گیری در هر نقطه انجام داد. که هر عنصر آرایه دارای وزن یکسان یک باشد. میتوان از کرنل $\begin{bmatrix} 1 & 2 & 1 \\ 1 & 2 & 1 \end{bmatrix}$ استفاده کرد. این کرنل تصویر را هموار میکند با وزن دادن به پیکسل های مرکزی.

در ابتدا برای درک بهتر این سوال من این دو کرنل را بر روی آرایه عددی سوال اول و تصویر داده شده در بخش عملی امتحان کردم و نتایج زیر حاصل شد:

```
first
[[ 0.  0.  0.  0.]
 [ 2. -1.  1. -6.]
 [-6.  3. -1.  1.]
 [ 0.  0.  0.  0.]]
second
[[16.  6.  4. 14.]
 [18.  5.  5.  8.]
 [14.  7.  5.  3.]
 [ 8. 10.  4.  4.]]
```

از خروجی عددی چیزی متوجه نمیشدم ولی نسبتاً تصاویر گویا بودند:

تصویر اول نتیجه ی اعمال کرنل $[-1, 0, 1]$ است و تصویر دوم نتیجه ی اعمال فیلتر $[[1], [2], [1]]$ است.



منابع مفید: [Sobel operator](#) و [Image Edge Detection using Sobel Operator from Scratch](#)

و کتاب مرجع

The kernels in Figs. 3.50(d) and (e) are called the *Sobel operators*. The idea behind using a weight value of 2 in the center coefficient is to achieve some smoothing by giving more importance to the center point (we will discuss this in more detail in Chapter 10). The coefficients in all the kernels in Fig. 3.50 sum to zero, so they would give a response of zero in areas of constant intensity, as expected of a derivative operator. As noted earlier, when an image is convolved with a kernel whose coefficients sum to zero, the elements of the resulting filtered image sum to zero also, so images convolved with the kernels in Fig. 3.50 will have negative values in general.

پاسخ سوال سوم

تصویر خروجی با $\text{ClipLimit} = 1$:

50	80	105	162
50	109	133	191
50	109	133	191

تصویر خروجی با $\text{ClipLimit} = 2$:

62	91	114	199
62	148	170	255
62	148	170	255

تمامی مراحل انجام هر یک در صفحات بعدی نوشته ام

پیاده سازی با OpenCV :

128	191	160	191
128	255	192	255
128	223	176	223

علت تفاوت جواب ها:

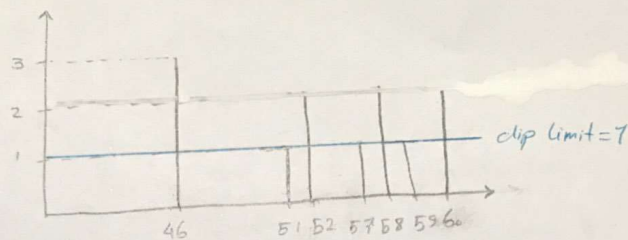
1- به دلیل اینکه openCV از REFLECT_101 استفاده میکند در حالی که در تبدیل دستی ما اینکار را نکردیم.

2- همچنین clahe از $\text{bilinear interpolation}$ استفاده میکند برای مرزهای تصاویر.

منبع:

[لینک 1](#) و [لینک 2](#)

46	51	57	59
46	52	58	60
46	52	58	60



این مقدار را به صورتی از اعداد در بازه 0 تا 255 انتخاب می کنیم.

$clip\ limit = 7 \Rightarrow 2 + 1 + 1 = 4 \rightarrow \frac{4}{256} = 0.015625$

$$\begin{aligned}
 52 &\rightarrow (L-1) \sum \frac{n_k}{n} = 255 \times \frac{3.828125}{9} \approx 108.4635 \xrightarrow{round} 108 \\
 46 &\rightarrow (L-1) \sum \frac{n_k}{n} = 255 \times \frac{1.734375}{9} \approx 49.14 \xrightarrow{round} 50 \\
 51 &\rightarrow (L-1) \sum \frac{n_k}{n} = 255 \times \frac{2.8125}{9} \approx 79.6875 \xrightarrow{round} 80 \\
 57 &\rightarrow (L-1) \sum \frac{n_k}{n} = 255 \times \frac{4.90625}{9} \approx 139.010417 \xrightarrow{round} 140 \\
 58 &\rightarrow (L-1) \sum \frac{n_k}{n} = 255 \times \frac{5.921875}{9} \approx 167.78 \xrightarrow{round} 168
 \end{aligned}$$

46	51	57	59
46	52	58	60
46	52	58	60

50	80	105	162
50	109	133	191
50	109	133	191

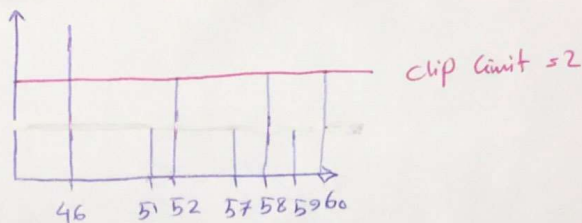
$$clip\ limit = 7 \Rightarrow 1 + 1 + 1 = 3 \rightarrow \frac{3}{256} = 0.01171875$$

$$\begin{aligned}
 51 &\rightarrow (L-1) \sum \frac{n_k}{n} = 255 \times \frac{1.609375}{9} = 45.59 \xrightarrow{round} 46 \\
 52 &\rightarrow (L-1) \sum \frac{n_k}{n} = 255 \times \frac{2.62109375}{9} = 74.2643 \xrightarrow{round} 75 \\
 57 &\rightarrow (L-1) \sum \frac{n_k}{n} = 255 \times \frac{4.25}{9} = 121.25 \xrightarrow{round} 121 \\
 58 &\rightarrow (L-1) \sum \frac{n_k}{n} = 255 \times \frac{5.9}{9} = 161.58 \xrightarrow{round} 162 \\
 59 &\rightarrow (L-1) \sum \frac{n_k}{n} = 255 \times \frac{7.5}{9} = 212.5 \xrightarrow{round} 212 \\
 60 &\rightarrow (L-1) \sum \frac{n_k}{n} = 255 \times \frac{9.1}{9} = 255.5 \xrightarrow{round} 255
 \end{aligned}$$

CS

Scanned with CamScanner

46	51	57	59
46	52	58	60
46	52	58	60



$$\text{clip limit} = 2 \rightarrow \frac{1}{256} = 0.00390625$$

$$46 \rightarrow (L-1) \sum \frac{1}{n} = 255 \times \frac{2.1835}{9} = 61.86 \approx \boxed{62}$$

$$51 \rightarrow = 255 \times \frac{3.203125}{9} = 90.75 \approx \boxed{91}$$

$$52 \rightarrow = 255 \times \frac{5.20703}{9} = 147.53 \approx \boxed{148}$$

$$57 \rightarrow = 255 \times \frac{6.2265}{9} = 176.41 \approx \boxed{177}$$

$$58 \rightarrow = 255 \times \frac{8.23046}{9} = 233.19 \approx \boxed{234}$$

46	51	57	59
46	52	58	60
46	52	58	60

62	91	114	199
62	148	170	255
62	148	170	255

$$\text{clip limit} = 2 \rightarrow \frac{0}{256} = 0$$

$$51 \rightarrow (L-1) \sum \frac{1}{n} = 255 \times \frac{1}{9} = 28.33 \approx \boxed{29}$$

$$52 \rightarrow = 255 \times \frac{3}{9} = \boxed{85}$$

$$57 \rightarrow = 255 \times \frac{4}{9} = 113.3 \approx \boxed{114}$$

$$58 \rightarrow = 255 \times \frac{6}{9} = \boxed{170}$$

$$59 \rightarrow = 255 \times \frac{7}{9} = 198.33 \approx \boxed{199}$$

$$60 \rightarrow = 255 \times \frac{9}{9} = \boxed{255}$$