==

What does sharpening do?

- Sharpening highlights transitions in intensity

How to enhance edges and other discontinuities (such as noise)?

- Differentiation.
- The strength of the response of a derivative operator is proportional to the magnitude of the intensity discontinuity at the point at which the operator is applied. Thus, image differentiation enhances edges and other discontinuities (such as noise) and de-emphasizes areas with slowly varying intensities.

Why use second derivatives in image sharpening?

- Edges in digital images often are ramp-like transitions in intensity, in which case the first derivative of the image would result in thick edges because the derivative is nonzero along a ramp. On the other hand, the second derivative would produce a double edge one pixel thick, separated by zeros. From this, we conclude that the second derivative enhances fine detail much better than the first derivative, a property ideally suited for sharpening images. Also, second derivatives require fewer operations to implement than first derivatives, so our initial attention is on the former.

==

Second derivative operator | The Laplacian operator

Definition: $$\Delta^2 f = \frac{\delta^2 f}{\delta x^2} + \frac{\delta^2 f}{\delta y^2}$$

For discrete 2D function:

$$\Delta^2 f(x, y) = f(x + 1, y) + f(x - 1, y) + f(x, y + 1) + f(x, y - 1) - 4f(x, y)$$

We can implement this equation using convolution with the kernel:

0	1	0
1	-4	1
0	1	0

Other laplacian kernels:

1	1	1
1	−8	1
1	1	1

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

Steps to sharpen an image with laplacian kernel

1. Filter the input with laplacian kernel
    - Convolution
    - The result may contain values < 0 or values > 255
    - Use clipping or scaling to display this filtered image.
2. If kernel center is non-negative add the filtered image to the input, else subtract it from input to get the sharpened image.
    - Subtracting two image may yield value in range [-255, 255]
    - Adding two image may yield value in range [0, 510]
    - Again use clipping or scaling to display the sharpened image.

Clipping:

If f(x, y) < 0, then f(x, y) = 0
If f(x, y) > 255, then f(x, y) = 255

Scaling:

Let's say we have an image g

$g_m = g - min(g)$ [element wise subtraction]

$g_s = K * (g_m / max(g_m))$ [element wise division]

g_s is our desired scaled image which contain values in range [0, K]