
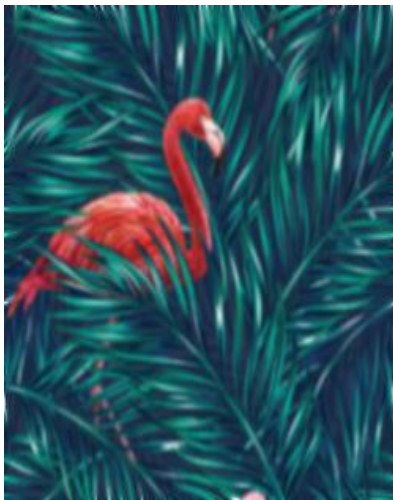
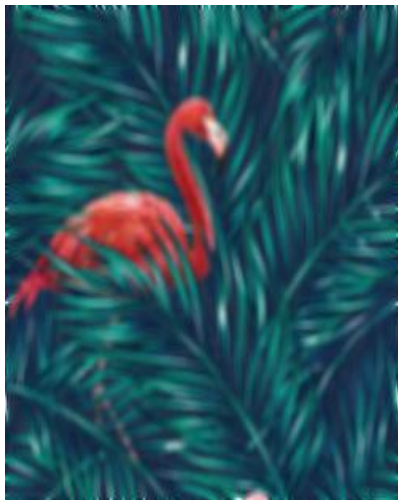


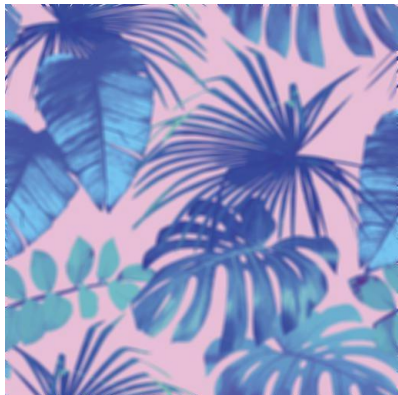


# Homework 1

Costa Huang ([sh3397@drexel.edu](mailto:sh3397@drexel.edu))

After passing the test cases, I ran the `hw1.py` on two images with varying parameters as shown below.

Original Image	Filtered image with $k = 5, \sigma = 1$	Filtered image with $k = 5, \sigma = 4$
		
Original Image	Filtered image with $k = 3, \sigma = 4$	Filtered image with $k = 7, \sigma = 4$
		

## Struggles

---

I personally had most of my troubles implementing the formula. First, the gaussian function at page 50 in the slide is probably incorrect. The formula is listed as

$$H[u, v] = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{\sigma^2}}$$

But the correct formula should be

$$H[u, v] = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}}$$

Second, the following convolution formula at page 60 was also somehow problematic.

$$H * F = \sum_{u=-k}^k \sum_{v=-k}^k H[u, v] F[i - u, j - v]$$

The issue is that  $H[-k, -k]$  is really referring to  $H[0, 0]$  in the implementation, but if you just write  $H[-k, -k]$  in the code, it will still run as python interprets it as the last  $k$  element in the array, which basically results in correlation instead of convolution. This is a subtle bug that was difficult to detect.

In any case, I eventually draw a picture and figured out what convolution actually means. It suggests you should do  $H[0, 0] * F[k, k] + H[0, 1] * F[k, k - 1] \dots$ , almost multiplying elements on a “circle” at opposite direction.

Written with [StackEdit](#).