Computer Vision HW0

An An Yu
5116372

2.

Fig.1 imshow

3.

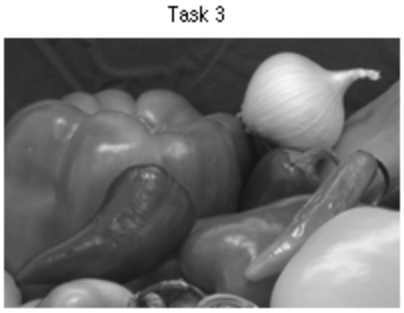

Fig.2 convert to grayscale with rbg2gray()

4.

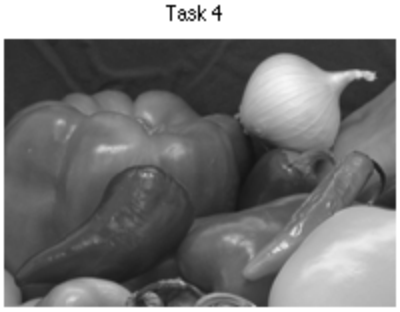

Fig.3 convert grayscale image to double using im2double()

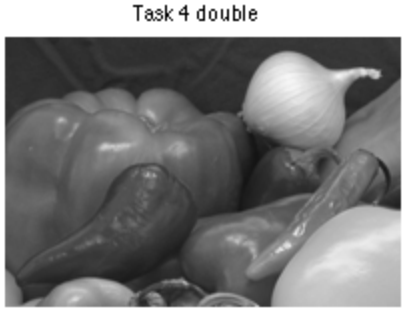

Fig.4 rescaled result after using double() to convert

- What is the difference between these two conversions?

im2double():
converts the intensity of our grayscale image to double precision and rescales the result to the range [0 1], which is the default display range for imshow().

double():
converts the grayscale image to double precision in range [0 255], which will display a blank image when using imshow.

So in order to preserve the image intensity, we need to rescale the image back to range [0 1] by dividing it with 255, which yields Fig.4.

- What is the difference between a double image and an unsigned integer image?

unsigned integers values are in a fixed range, eg: [0 128], [0 255], and each pixel can only adopt one integer value as its intensity level.

double images typically range from 0 to 1, but the number of values each pixel can adopt would be the same as the double precision.

5.

Implementation: Extracting submatrix from image by matrix indexing

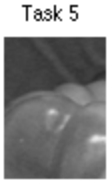

Fig.5 obtained by Crop(img,5,5,50,70)

6.

Implementation: use nested for-loops to implement 2D convolution and convolve the image intensity with a nxn averaging window (all values are 1s).

- Window size = 3:

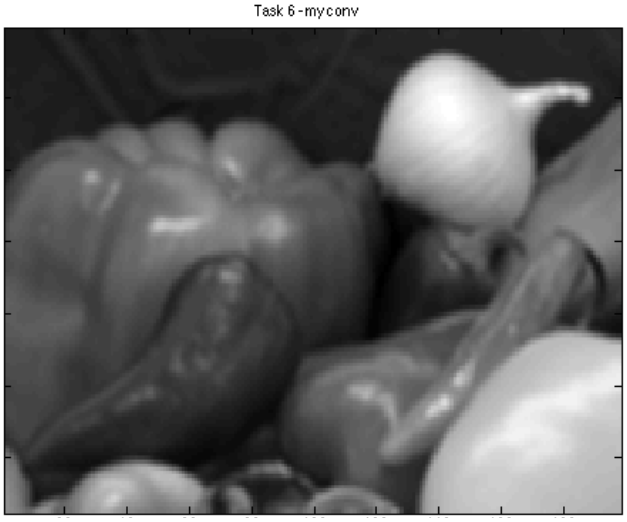

Fig.6 my implementation using smaller windows on pixels that have fewer then nxn-1 neighbors

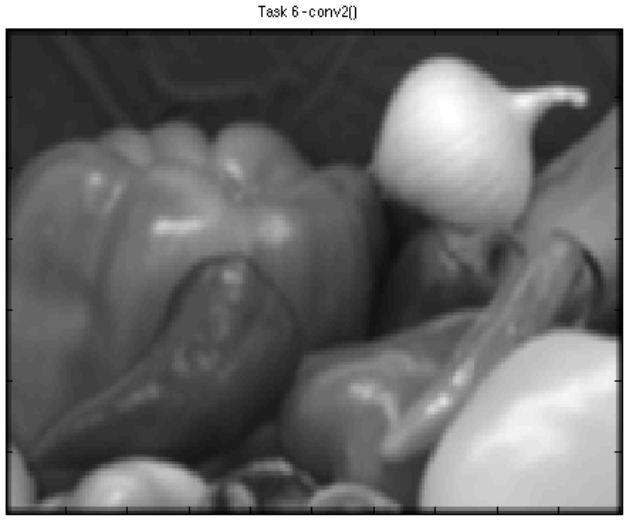

Fig.7 using conv2() zero padding results in dark borders

- Window size = 15

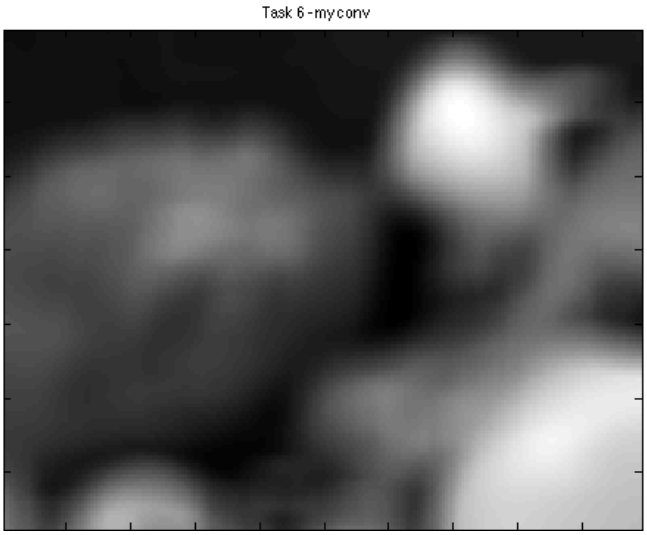

Fig.8 my implementation

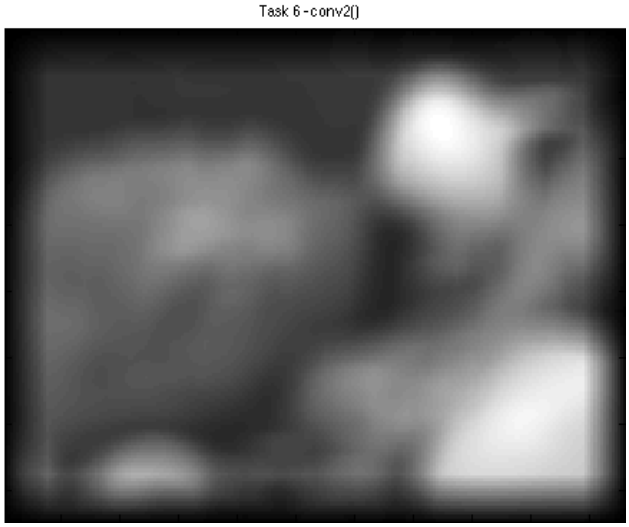

Fig.9 using conv2() the effect of zero-padding increases with window size