EE568 - Digital Image Processing, Winter 2021

Assignment 5

**Due Time/Date: 8:30** **am, Tuesday, 2/9/2021**

*Please follow submission guidelines.*

* Solve the following problems using your own Matlab programs. Unless specified in the problem, do not use advanced MATLAB and Python image editing commands such as *imadjust*, *medfilt2, histeq, edge,* cv2.GaussianBlur, cv2.medianBlur etc... .

1. Implementing image high boost filtering with matrix multiplication:

To take advantage of existing high-performance linear algebra libraries (for example, BLAS: basic linear algebra subprograms), convolution could be implemented using matrix-matrix multiplications. For example, the multiplication of a 3 by 3 image patch with the coefficients of a 3 by 3 high boost filter can be implemented with a vector dot product:

For an image larger than 3 by 3, we could extract a 3 by 3 patch around each pixel (for this question, we do not need to consider patches that have elements outside of image boundary). Each 3 by 3 patch could be flattened as a 9-element vector. For example, we could extract four patches from a 4 by 4 image, those four patches are: patch around pixel : ; patch around pixel : ; patch around pixel : and patch around pixel : .

The convolution results at :

is equivalent to:

(1)

Then reshape to to get the 2D filtered image.

Write a script to apply the high boost filter on **5\_1.bmp** using the matrix multiplication.

Hint:

1. Please do not use for loops to go over every pixel, the filtering process must have the format of matrix multiplication: for MATALB: C=A\*B; for numpy python: C=numpy.matmul(A,B).

2. Image 4\_2 have RGB three channels. We could process those three channels independently.

3. Useful functions: MATLAB: cat, reshape; Python: numpy.stack, reshape; and some index slicing tricks. Use some slicing tricks and stack to extract image patches into shape (height-2, width-2, 9), then reshape this 3D array into matrix ((height-2)\*(width-2), 9), this is the left matrix of equation (1). Apply equation (1), then reshape the result from ((height-2)\*(width-2),1) to (height-2, width-2), this is the filtered result.

4. We may need to set negative pixel values to 0, and pixel values larger than 1.0 (or 255) to 1.0 (or 255).

1. Sharpen **5\_1.bmp** using a high boost filter. You can use the 2D convolution commands in MATLAB or Python.