Assignment3 / {Kernels}

Graphics Programming / Tristan Goodell

Box Blur / {pic_1_2_x}

• Task: Develop a 3x3, 7x7, 1x7, and 1x3 Box Blur.



image1.png Original



pic_1_2_1.png 3x3



pic_1_2_2.png 7x7



pic_1_2_3.png 1x7



pic_1_2_4.png 1x3

Box Blur / {kernels}

```
box3x3kernel=np.float64([[1/9,1/9,1/9],
                          [1/9, 1/9, 1/9],
                          [1/9,1/9,1/9]]
box7x7kernel=np.float64([[1/49,1/49,1/49,1/49,1/49,1/49,1/49],
                          [1/49, 1/49, 1/49, 1/49, 1/49, 1/49, 1/49],
                          [1/49,1/49,1/49,1/49,1/49,1/49]
                          [1/49, 1/49, 1/49, 1/49, 1/49, 1/49, 1/49],
                          [1/49, 1/49, 1/49, 1/49, 1/49, 1/49, 1/49],
                          [1/49, 1/49, 1/49, 1/49, 1/49, 1/49, 1/49],
                          [1/49, 1/49, 1/49, 1/49, 1/49, 1/49, 1/49]]
box1x7kernel=np.float64([[1/7,1/7,1/7,1/7,1/7,1/7,1/7]])
box1x3kernel=np.float64([[1/3,1/3,1/3]])
```

Box Edge Effects / {pic_1_3_x}

• Task: Using the 7x7 kernel try out all edge effects.



image1.png
Original



pic_1_3_1.png
Reflect



pic_1_3_2.png
Constant



pic_1_3_3.png Replicate



pic_1_3_4.png Replicate101

Box Edge Effects / {code}

box7x7kernel=np.float64([[1/49,1/49,1/49,1/49,1/49,1/49,1/49],

```
[1/49,1/49,1/49,1/49,1/49,1/49],
[1/49,1/49,1/49,1/49,1/49,1/49],
[1/49,1/49,1/49,1/49,1/49,1/49],
[1/49,1/49,1/49,1/49,1/49,1/49],
[1/49,1/49,1/49,1/49,1/49,1/49],
[1/49,1/49,1/49,1/49,1/49],
[1/49,1/49,1/49,1/49,1/49]])

pic_1_3_1=normalize(cv2.filter2D(img,cv2.CV_64F,box7x7kernel,borderType=cv2.BORDER_REFLECT))
pic_1_3_2=normalize(cv2.filter2D(img,cv2.CV_64F,box7x7kernel,borderType=cv2.BORDER_CONSTANT))
pic_1_3_3=normalize(cv2.filter2D(img,cv2.CV_64F,box7x7kernel,borderType=cv2.BORDER_REPLICATE))
pic_1_3_4=normalize(cv2.filter2D(img,cv2.CV_64F,box7x7kernel,borderType=cv2.BORDER_REFLECT101)
```

Box Edge Diff / {pic_1_4}

• Task: Apply the 1x7 and its transposed. Do a diff with 7x7.



image1.png Original



pic_1_4 Diff

Diff Stats:

Max: 100

Min: 0

Gaussian Blur / {pic_2_2_x}

• Task: Develop a 3x3, 7x7, 1x7, and 1x3 Gaussian Blur.



image1.png Original



pic_2_2_1.png 3x3



pic_2_2_2.png 7x7



pic_2_2_3.png 1x7



pic_2_2_4.png 1x3

Gaussian Blur / {kernels}

```
gauss3x3kernel=np.float64([[0.0625,0.125,0.0625],
                            [0.125, 0.25, 0.125],
                            [0.0625,0.125,0.0625]])
gauss7x7kernel=np.float64([[0.00097656,0.00341797,0.00683594,0.00878906,0.00683594,0.00341797,0.00097656],
                            [0.00341797, 0.01196289, 0.02392578, 0.03076172, 0.02392578, 0.01196289, 0.00341797]
                            [0.00683594, 0.02392578, 0.04785156, 0.06152344, 0.04785156, 0.02392578, 0.00683594],
                            [0.00878906, 0.03076172, 0.06152344, 0.07910156, 0.06152344, 0.03076172, 0.00878906],
                            [0.00683594, 0.02392578, 0.04785156, 0.06152344, 0.04785156, 0.02392578, 0.00683594]
                            [0.00341797, 0.01196289, 0.02392578, 0.03076172, 0.02392578, 0.01196289, 0.00341797],
                           [0.00097656, 0.00341797, 0.00683594, 0.00878906, 0.00683594, 0.00341797, 0.00097656]])
gauss1x7kernel=np.float64([[0.03125,0.109375,0.21875,0.28125,0.21875,0.109375,0.03125]])
gauss1x3kernel=np.float64([[0.25,0.5,0.25]])
```

Gauss Edge Effects / {pic_2_3_x}

• Task: Using the 7x7 kernel try out all edge effects.



image1.png Original



pic_2_3_1.png Reflect



pic_2_3_2.png Constant



pic_2_3_3.png Replicate



pic_2_3_4.png Replicate101

Gauss Edge Effects / {code}

Gauss Edge Diff / {pic_2_4}

• Task: Apply the 1x7 and its transposed. Do a diff with 7x7.



image1.png Original



pic_2_4 Diff

Diff Stats:

Max: 75 Min: 0

Sharpen / {pic_3_2_x}

• Task: Make a 3x3, 5x5, 7x7, and 9x9 sharpen kernel.



image1.png Original



pic_3_2_1.png 3x3



pic_3_2_2.png 5x5



pic_3_2_3.png 7x7



pic_3_2_4.png 9x9

Sharpen / {kernels}

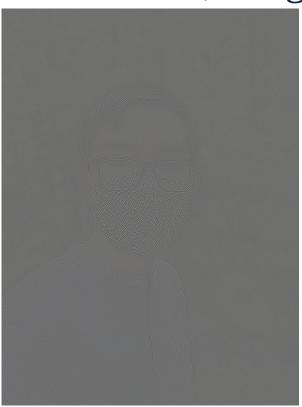
```
sharpen3x3kernel=np.float64([[0,-1,0],
                                [-1,18,-1].
                                [0, -1, 0]
sharpen5x5kernel=np.float64([[0,0,-1,0,0],
                                [0,-1,-2,-1,0],
                                [-1, -2, 50, -2, -1],
                                [0,-1,-2,-1,0],
                                [0,0,-1,0,0]]
sharpen7x7kernel=np.float64([[0,0,-1,-2,-1,0,0],
                                [0, -1, -2, -3, -2, -1, 0],
                                [-1, -2, -3, -4, -3, -2, -1],
                                [-2, -3, -4, 98, -4, -3, -2],
                                [-1, -2, -3, -4, -3, -2, -1],
                                [0,-1,-2,-3,-2,-1,0],
                                [0,0,-1,-2,-1,0,0]
sharpen9x9kernel=np.float64([[0,0,-1,-2,-3,-2,-1,0,0],
                                [0,-1,-2,-3,-4,-3,-2,-1,0],
                                [-1, -2, -3, -4, -5, -4, -3, -2, -1],
                                [-2, -3, -4, -5, 162, -5, -4, -3, -2],
                                [-1, -2, -3, -4, -5, -4, -3, -2, -1],
                                [0,-1,-2,-3,-4,-3,-2,-1,0],
                                [0,0,-1,-2,-3,-2,-1,0,0]])
```

Sharpen 10

• If we apply 5x5 Sharpen kernel ten times, we get:



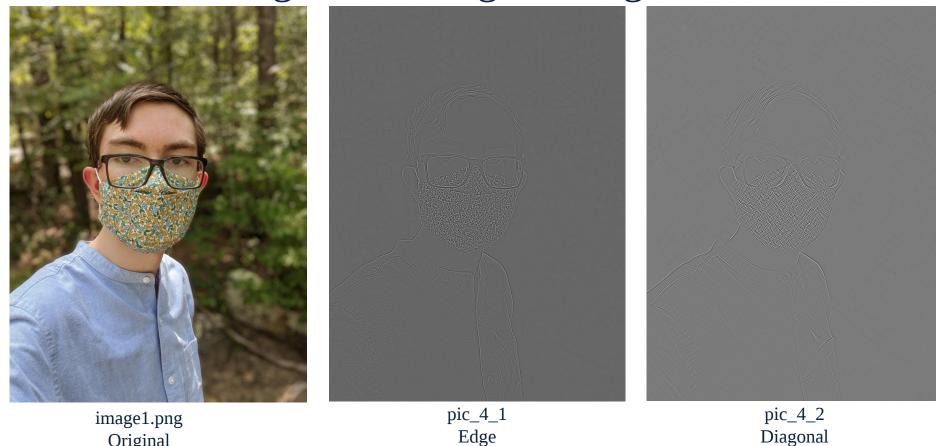
image1.png Original



superSharpen.png
5x5 Sharpen * 10

Edge Detection / {pic_4_x}

• Task: Make an edge and a diagonal edge kernel to detect corners.



Original

Edge

Edge Detection / {kernels}