6.1 Custom filter in adobe photoshop is not strictly linear filter because number of points, color to render, calculating boundaries & edges and pixelating RGB values would not be super accurate.

A piece of paper with writing on it

Description automatically generatedA piece of paper with writing on it

Description automatically generated with medium confidence

[Q II-1 - 30pts]

- Implement an ImageJ plugin that takes a gray-level image and produce bright “whitish” highlights on the edges. You need to make sure that the range of any pixel does not exceed 255. Test the code on the image ireland-03gray.tif

import numpy as np

import cv2

from matplotlib import pyplot as plt

irelandimg = cv2.imread('D:/CS334/ireland-03gray.tif', 1)

plt.figure(figsize = (10, 10))

plt.imshow(irelandimg)

plt.title("Original image")

#Plot Original image

plt.show()

# convert to grayscale

gray = cv2.cvtColor(irelandimg, cv2.COLOR\_RGB2GRAY)

# create a binary thresholded image

\_, binary = cv2.threshold(gray, 225, 255, cv2.THRESH\_BINARY\_INV)

plt.figure(figsize = (10, 10))

plt.imshow(binary, cmap="gray")

plt.title("Whitish highlights")

plt.show()

print('Size of Image: {}'.format(irelandimg.size))

print('Maximum RGB value: {}'.format(irelandimg.max()))

print('Minimum RGB value: {}'.format(irelandimg.min()))

A picture containing text, old

Description automatically generatedOutput:

- Modify the plugin such that it takes an input-colored image and produce the highlights on the edges. You need first to obtain a gray-scale image by averaging the three-color channels, and then perform the gradient operation on that image. The scaled magnitude of the image gradient is then added to each of the color channels. Test the code on the provided image. Test the code on the image Amsterdam.JPG

image = cv2.imread('D:/CS334/ireland-03gray.tif', 0)

sobelx = cv2.Sobel(fiximgage,cv2.CV\_64F,1,0,ksize=5)

fiximgage = cv2.cvtColor(image, cv2.COLOR\_BGR2RGB)

#extract channels

R, G, B = fiximgage[:,:,0], fiximgage[:,:,1],fiximgage[:,:,2]

#formula

channel = (R+B+G) / 3

print(channel)

plt.figure(figsize = (10, 10))

plt.imshow(channel, cmap='gray')

plt.imshow(sobelx)

plt.show()

Output: A picture containing text

Description automatically generated

Modify the plugin such that it produces dark highlights on the edges. Test the code on the image Amsterdam.JPG

image = cv2.imread('D:/CS334/Amsterdam.JPG')

sobelx = cv2.Sobel(fiximgage,cv2.CV\_64F,1,0,ksize=5)

fiximgage = cv2.cvtColor(image, cv2.COLOR\_BGR2RGB)

#extract channels

R, G, B = fiximgage[:,:,0], fiximgage[:,:,1],fiximgage[:,:,2]

#formula

channel = (R+B+G) / 3

print(channel)

plt.figure(figsize = (10, 10))

plt.imshow(channel, cmap='gray')

plt.imshow(sobelx)

plt.show()

Output:

A picture containing diagram

Description automatically generated

[Q II-2 – 40pts]

Point operations on two images can be used to detect moving objects by subtracting consecutive images in an image sequence. In this problem we will use a sequence of operations to detect soccer player in a field from two frames. Use the images soccer1, soccer2 for this problem. In general, we call them images I1, I2. Use ImageJ to implement the following steps:

1. Given two images I1, I2, use ImageJ point operation to obtain the absolute difference image d=abs(I1-I2).

import cv2 as cv

l1 = cv2.imread('D:/CS334/soccer1.bmp')

plt.figure(figsize = (10, 10))

plt.imshow(l1)

plt.title("Soccor 1")

l2 = cv2.imread('D:/CS334/soccer2.bmp')

plt.figure(figsize = (10, 10))

plt.imshow(l2)

plt.title("Soccor 2")

#calculating absolute differences

imaged = cv.absdiff(l1, l2)

plt.figure(figsize = (10, 10))

#display image computed of the image

plt.imshow(imaged)

plt.title("absolute difference of image")

Output:

A screenshot of a video game

Description automatically generated with medium confidenceGraphical user interface, diagram

Description automatically generated

1. Plot the histogram for the absolute difference image d.

plt.hist(imaged.ravel())

Output:

Chart

Description automatically generated

3- Visualize the absolute difference image d. To get a good visualization, you need to stretch the range of values in d to be from 0-255. Use a point operation to achieve this stretching. Show the result.

#Visualize the absolute difference imaged. To get a good visualization, you need to stretch the range of values in d to

#be from 0-255. Use a point operation to achieve this stretching. Show the result

plt.figure(figsize = (10, 10))

plt.hist(imaged.ravel(), 256, [0, 256])

#Maximum intensity Level

Intense = imaged.max()

#Calulate each intensity level

TotalIntensity = Intense - imaged

plt.figure(figsize = (10, 10))

plt.imshow(imaged)

plt.imshow(TotalIntensity)

Chart

Description automatically generatedA group of people walking in the snow

Description automatically generated with low confidenceOutput:

[Q II-4 – 15pts] Design a linear filter that creates a horizontal blur over a length of 7 pixels, thus simulating the effect of camera movement during exposure. Implement this horizontal blur filter and test your code on the image 100.tif as well as your selfi image from the previous assignment

filteredimage = cv2.imread('D:/CS334/100.tif', 1)

size = 7

kernel = np.zeros((size, size))

kernel[int((size - 1) / 2), :] = np.ones(size)

kernel /= size

horizonal = cv2.filter2D(filteredimage, -1, kernel)

plt.figure(figsize = (10, 10))

plt.imshow(filteredimage)

plt.title("Original")

plt.figure(figsize = (10, 10))

plt.imshow(horizonal)

plt.title("Blur")

A picture containing text, person, indoor

Description automatically generatedOutput:

#testing on selfie

selfie = cv2.imread('C:/Users/sanji/Desktop/sanjit.jpg', 1)

fiximgage = cv2.cvtColor(selfie, cv2.COLOR\_RGB2BGR)

size = 7

kernel = np.zeros((size, size))

kernel[int((size - 1) / 2), :] = np.ones(size)

kernel /= size

horizonal = cv2.filter2D(fiximgage, -1, kernel)

plt.figure(figsize = (10, 10))

plt.imshow(fiximgage)

plt.title("Original")

plt.figure(figsize = (10, 10))

plt.imshow(horizonal)

plt.title("Blur")

A person holding a water bottle

Description automatically generated with medium confidenceA person holding a water bottle

Description automatically generated with medium confidenceOutput: