Abubakar Kazini
L220056

Assignment 2 - Image processing

Q.1 part 1: salt and pepper

Start:

function to add salt and pepper (Image, probability of salt, probability of pepper):

inputs:

Image : a grey scale image

probability of salt : white pixel

probability of pepper : black pixel

total probability = prob salt + prob pepper

Salt_ratio = prob salt / total probability

for loop:

for each pixel in the image:

generate a random value b/w 0 and 1

if random value < total probability:

if random value < salt ratio * total probability:

Set pixel intensity to max value (255) * salt

else:

Set pixel intensity to min value (0) * Pepper (for 8 bit img)

Return the noisy image

finish:

part 2 Q1:

start:

function to apply Convolutional filter(Image, Kernel, padding):

Inputs:

Image: the noisy image (2D array)

Kernel: value size of the filter (3)

Padding: string padding type.

step 1: set the Kernel size

if kernel = false, set it to default (3)

step 2: apply the filter

use scipy's median_filter function with (kernel, & padding)

step 3: return the filtered image after filtering

the main program

1- input the noisy image as grayscale

2- call the filter function (with noisy image, Kernel = 3)

3- plot it if you want to visualize it.

finish:

Q2: (RGB)

Start:

Function to enhance RGB contrast(Image):

Inputs:

Image: 3D RBG Image.

step 1: Split the Image into 3 Rgb channels:

R = image[:, :, 0] ← Red channel

G = Image[:, :, 1] ← Green channel

B = image[:, :, 2] ← Blue channel

Step 2: define the function to apply linear Stretching (channel)

find the min and max pixel value in the channel:

minval = min value of the channel

maxval = max value of the channel

apply the linear Stretching formular to pixel:

$$P_{adjusted} = (P - P_{min}) \left(\frac{255}{P_{max} - P_{min}} \right)$$

P ← channel

return adjusted content as integer (8 bit) uint8.

step 3: apply linear stretching function to each channel

R = ls function (R)

G = ls function (g)

b = ls function (b)

Step 4: Combine the adjusted channel back to RGB image

enhanced_image = stack R, G, B into a single 3D array

Return enhancedImage

Stop:

Q 2B HSL:

Start :

function for hsl (Image , adjustmentfactor):

Inputs:

Image = RGB Image

adjustment_factor = floating value for scaling the enhance

Step 1: Convert RGB to HSL

hsl_img = Convert_rgb_hsl_function (Image)

Step 2: extract the lightness channel from hsl image

l_channel = hsl_img [:, :, 2]

Step 3: enhance contrast by adjusting the lightness channel.

L_channel = l_channel * adjustment factor

clip the channel value to the rang (0, 1]

Step 4: Update the hsl image with the adjusted light channel

hsl_image [:, :, 2] = L_channel

Step 5 : Convert the hsl back to RGB

rgb_image = Convert_hsl_rgb_function (hsl_image)

Return rgb_image

Stop.