Week 5 Homework Q1

Telmen Enkhbold

San Fransico Bay University

CE480 - Java and Internet Application

Dr. Chang, Henry

10/12/2023

# Author Note

# The Question

Noise Reduction

Step 1: Study the Pattern Recognition Process

Step 2: Study Pattern Recognition - implementing your code without using an image processing library.

Step 3: Study the Concept of Blurring

Step 4: Study the Concept of Mask (Filters)

Step 3: Using a Gausian filter to remove several salt and pepper noises.

Step 3.1: Asking ChatGPT to provide sample Java code.

You can get a hint from ChatGPT.

Step 3.2: Using the "Noise Image" and "Clean Image" shown on Concept of Blurring to test your program.  
  
import java.awt.image.BufferedImage;

import java.io.File;

import java.util.Random;

import javax.imageio.ImageIO;

public class NoisyImageProcessor {

    public static void main(String[] args) throws Exception {

        // Generate a random noisy image

        int width = 512;  // You can adjust this

        int height = 512; // You can adjust this

        BufferedImage generatedNoisyImage = generateNoisyImage(width, height);

        // Add additional noise and then denoise it using Gaussian filter

        BufferedImage noisyImage = addNoise(generatedNoisyImage, 25);

        BufferedImage denoisedImage = denoiseUsingGaussianFilter(noisyImage);

        // Save the images

        ImageIO.write(noisyImage, "jpg", new File("path\_to\_save\_noisy\_image.jpg"));

        ImageIO.write(denoisedImage, "jpg", new File("path\_to\_save\_denoised\_image.jpg"));

    }

    public static BufferedImage generateNoisyImage(int width, int height) {

        Random random = new Random();

        BufferedImage noisyImage = new BufferedImage(width, height, BufferedImage.TYPE\_INT\_RGB);

        for (int y = 0; y < height; y++) {

            for (int x = 0; x < width; x++) {

                int red = random.nextInt(256);

                int green = random.nextInt(256);

                int blue = random.nextInt(256);

                int rgb = (red << 16) | (green << 8) | blue;

                noisyImage.setRGB(x, y, rgb);

            }

        }

        return noisyImage;

    }

    public static BufferedImage addNoise(BufferedImage image, int strength) {

        Random random = new Random();

        int width = image.getWidth();

        int height = image.getHeight();

        BufferedImage noisyImage = new BufferedImage(width, height, image.getType());

        for (int y = 0; y < height; y++) {

            for (int x = 0; x < width; x++) {

                int rgb = image.getRGB(x, y);

                int noise = (int) (strength \* (random.nextFloat() - 0.5));

                int newRed = clamp((rgb >> 16) & 0xFF + noise, 0, 255);

                int newGreen = clamp((rgb >> 8) & 0xFF + noise, 0, 255);

                int newBlue = clamp(rgb & 0xFF + noise, 0, 255);

                noisyImage.setRGB(x, y, (newRed << 16) | (newGreen << 8) | newBlue);

            }

        }

        return noisyImage;

    }

    public static BufferedImage denoiseUsingGaussianFilter(BufferedImage image) {

        return image;

        // ... [Same denoiseUsingGaussianFilter method as provided earlier]

    }

    public static int clamp(int value, int min, int max) {

        return Math.max(min, Math.min(max, value));

    }

}