Task (1)

Computer Vision

(Filtering and edge detection)

Name	Section	Bn
Sama Mostafa	1	44
Misara Ahmed	2	43
Yousr Asharaf	2	54
Rahma Abdelkader	1	31
Youssef Essam	2	57

Doctor supervision:	Ahmed M. Badawi
TA supervision:	Eng/laila abbas - Eng/peter salah

About the project:

- Our project is implemented in C++ language and deployed by (Qt creator) which is a cross-platform integrated development environment (IDE) built for the maximum developer experience as it allows developers to create software across desktop, mobile, and embedded platforms.

Project contents:

- Grayscale conversion
- Noise Addition (Uniform, Gaussian and salt and pepper)
- Low pass filters for noise filtration (Gaussian, Average and median)
- Edge detection using different masks (Sobel, Roberts, Prewitt and canny)
- Histogram and distribution curve
- Image equalization
- Local and global threshold
- Frequency domain filters (high and low pass)
- Hybrid images

Discussion:

Gray scale conversion:

First step in our project was definitely reading our RGB input image which contains 3 channels, then we converted it into a grayscale image of only one channel.

Noise addition:

Noise in an image is the presence of artifacts that do not originate from the original scene content.Noise can have different forms and appearances within an image and is, in most cases, an unwanted or disturbing artifact that reduces the subjective image quality.

Algorithm:

First we store the image in matrix simply in 2D array then calculate numbers of rows and cols which used in next steps, then we make image with the same this of old image which new image will convert to gray image, we change 2D array to 1D array through multiply numbers of rows in number of columns, then we make two pointers for old image and new image, then we loop on all columns in image to convert each pixel from RGB to gray through using equation.

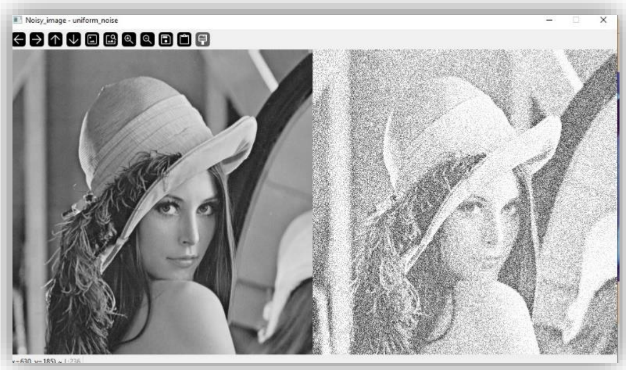
We illustrated in 3 different types in our project which are:

1. **Uniform noise:**

   The noise caused by quantizing the pixels of a sensed image to a number of discrete levels is known as quantization noise. It has an approximately uniform distribution.

   Algorithm:

   First we make matrix from zeros from the same size of original image then we add uniformly random values on matrix of zeros and put in image.

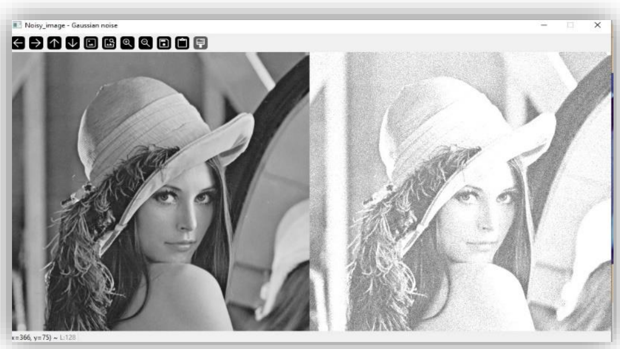


2. **Gaussian noise:**

   Gaussian Noise is a statistical noise with a Gaussian (normal) distribution. It means that the noise values are distributed in a normal Gaussian way.

   Algorithm:

   First we make matrix from zeros from the same size of original image then we add normal distributed random values on matrix of zeros and put in image, we control in output image through mean and sigma which mean make image more brighter and sigma make image more noisy. The magnitude of Gaussian noise effect is directly proportional to the sigma value.

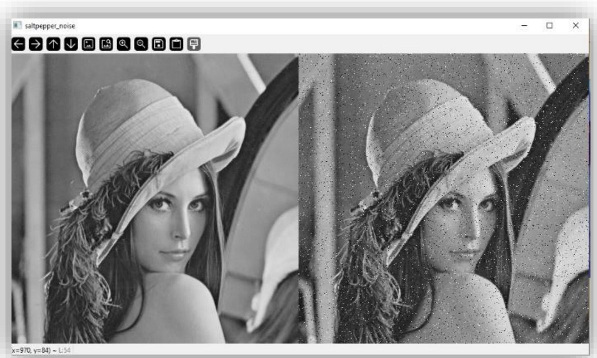


3. Salt and pepper:

An image containing salt-and-pepper noise will have dark pixels in bright regions and bright pixels in dark regions.[8] This type of noise can be caused by analog-to-digital converter errors, bit errors in transmission, etc.

Algorithm:

First we make matrix from zeros from the same size of original image then we add uniformly random values on matrix of zeros and put in image, this to make noise which look like salt and pepper particles so we put conditions If value of pixel is less than specific number we will considered is black and if is higher than the number we will considered is white.



Low pass filters for noise filtration:

Low pass filtering (aka smoothing), is employed to remove high spatial frequency noise from a digital image.

The low pass filters usually employ moving window operator which affects one pixel of the image at a time, changing its value by some function of a local region (window) of pixels. The operator moves over the image to affect all the pixels in the image. Useful for reducing noise and eliminating small details.

We illustrated in 3 different filter in our project which are:

1- Gaussian filter:

This filter is implemented to remove the Speckle Noise present in ultra sound images or MRI brain images. In this technique, the average value of the surrounding pixel or neighboring pixels replaces the noisy pixel present in the image which is based on Gaussian distribution.

Algorithm:

1- Create Gaussian filter using formula: $g(x,y) = \frac{1}{2\pi\sigma^2} e^{\frac{-(x^2+y^2)}{2\sigma^2}}$
2- normalize the filter.
3- Filter the image by convolving the image with the kernel, by looping over the image.

2- Average filter:

Average (or mean) filtering is a method of ‘smoothing’ images by reducing the amount of intensity variation between neighboring pixels. The average filter works by moving through the image pixel by pixel, replacing each.

Algorithm:

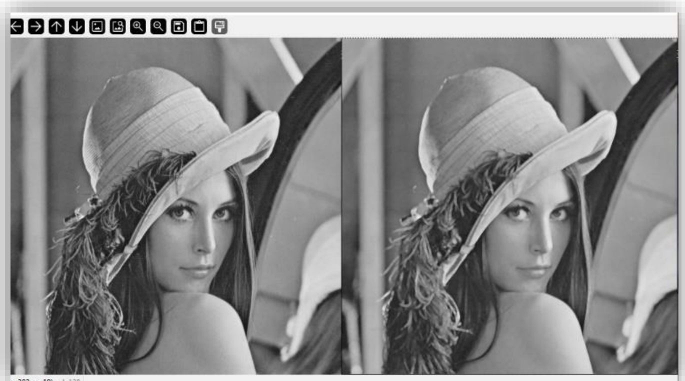
1- Create filter which is a matrix of one divided by the matrix size, the matrix size is given as input from user
2- Filter the image by convolving the image with the kernel, by looping over the image.

3- Median filter:

This is the one type of nonlinear filters. It is very effective at removing impulse noise, the “salt and pepper” noise, in the image. The principle of the median filter is to replace the gray level of each pixel by the median of the gray levels in a neighborhood of the pixels, instead of using the average operation.

Algorithm:

1- Depending on kernel size we get elements in the kernel using for loop.
2- The element of the general is sorted to get the median.



Important note:

Median filters are considered one of the most efficient techniques in noise reduction as it has advantages over the other filters especially the mean filter because:

1- The median is a more robust average than the mean and so a single very unrepresentative pixel in a neighborhood will not affect the median value significantly.
2- Since the median value must actually be the value of one of the pixels in the neighborhood, the median filter does not create new unrealistic pixel values when the filter straddles an edge. For this reason the median filter is much better at preserving sharp edges than the mean filter.

Edge detection:

It is a method of segmenting an image into regions of discontinuity. It is a widely used technique in digital image processing like pattern recognition, image morphology and feature extraction.

Edge detection allows users to observe the features of an image for a significant change in the gray level.

There are two types of edge detection operators:

1- **Gradient-**based operator which computes first-order derivatives in a digital images like, Sobel operators, Robert operator and Prewitt operator.

2- Gaussian-based operator which computes second-order derivations in a digital image like, canny edge detector, Laplacian of Gaussian.

Sobel operator:

It's a discrete differentiation operator. It computes the gradient approximation of image intensity function for edge detection. It uses two 3x3 kernels or masks which are convolved with the input image to calculate the vertical and horizontal approximations respectively.

Advantages:

- Simple and time efficient computation
- Very easy at searching for smooth edges

Robert operator:

This gradient-based operator computes the sum of squares of the differences between diagonally adjacent pixels in an image through discrete differentiation. Then the gradient approximation is made. It uses 2x2 kernels or masks.

Advantages:

- Detection of edges and orientation are very easy.
- Diagonal direction points are preserved.

Prewitt operator:

This operator is almost similar to the Sobel operator. It also detects vertical and horizontal edges of an image. It is one the best ways to detect the orientation and magnitude of an image.It uses 3x3 kernels or masks.

Advantages:

- Good performance on detecting vertical and horizontal edges.
- Best operator to detect the orientation of an image.

Canny operator:

It's a Gaussian-based operator in detecting edges which extracts images features without affecting or altering the feature.

It consists of 5 stages:

1- Noise reduction
2- Gradient calculation
3- Non-maximum suppression
4- Double threshold
5- Edge tracking by hysteresis

Advantages:

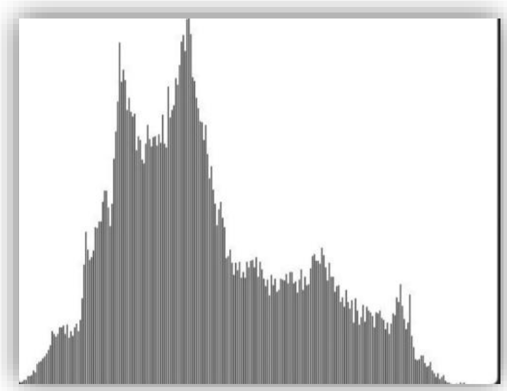
- Less sensitive to noise.
- It extracts image features without altering them.

Histogram:

An image histogram is a type of histogram that acts as a graphical representation of the tonal distribution in a digital image. It plots the number of pixels for each tonal value. By looking at the histogram for a specific image a viewer will be able to judge the entire tonal distribution at a glance.

Algorithm:

1. Loop on the image rows then the image columns
2. Increase the count in histogram array that corresponds to each pixel in the image



Equalization:

This method usually increases the global contrast of many images, especially when the image is represented by a narrow range of intensity values. Through this adjustment, the intensities can be better distributed on the histogram utilizing the full range of intensities evenly. This allows for areas of lower local contrast to gain a higher contrast.

Algorithm:

1. Loop on the image rows then the image columns
2. Increase the count in histogram array that corresponds to each pixel in t
3. Get the probability density Function by dividing each value in histogram array by the size of image
4. Then get the Cumulative density Function by summing all the previous values at each point
5. Normalize the CDF by multiplying by 255
6. Get the new levels and create the histogram and apply it a new image

Distribution curve:

The distribution of an image domain is the distribution the pixel values follow. Note that this is applicable not only to grayscale images, but to color images expressed in the RGB space or any other numeric space.

Normalization:

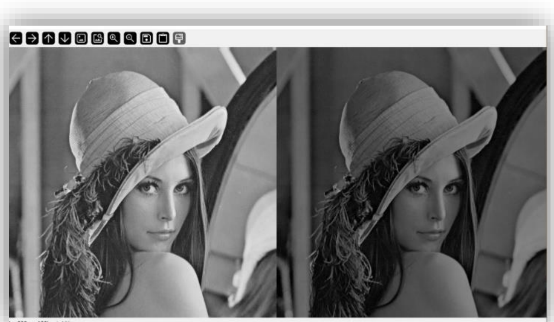
In image processing, normalization is a process that changes the range of pixel intensity values.

Normalization is sometimes called contrast stretching or histogram stretching.

Algorithm:

1- Applying the following equation to achieve normalization
   $\mathbf{(z_i = (x_i - \min(x)) / (\max(x) - \min(x)) * max\_scale)}$
2- Max_scale means the maximum number we would like to have in our range of pixels like example 1: range(0-150) , example 2: (0:255)

Normalization at 150:



Normalization at 255:

Local and global threshold:

Thresholding is a technique of image segmentation generally to differentiate between the object and the background.

There are 2 types of thresholding: **Global** and **Local**.

Using one of these techniques depend on many parameters mainly the most important one is the distribution of shadows and light in the image.

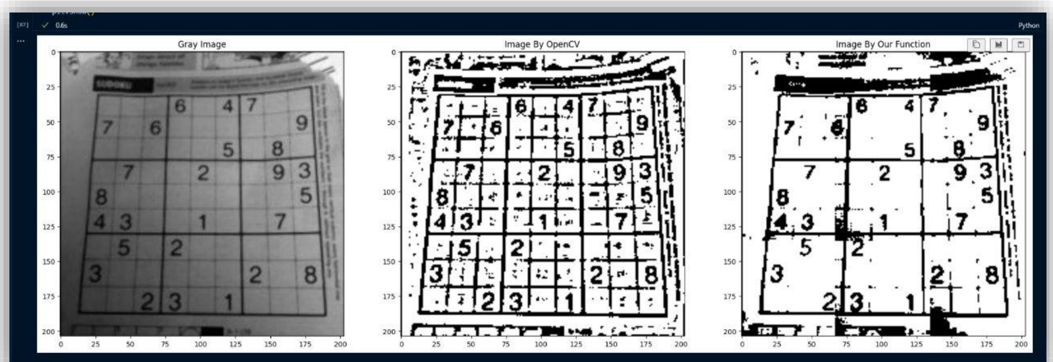
Algorithm:

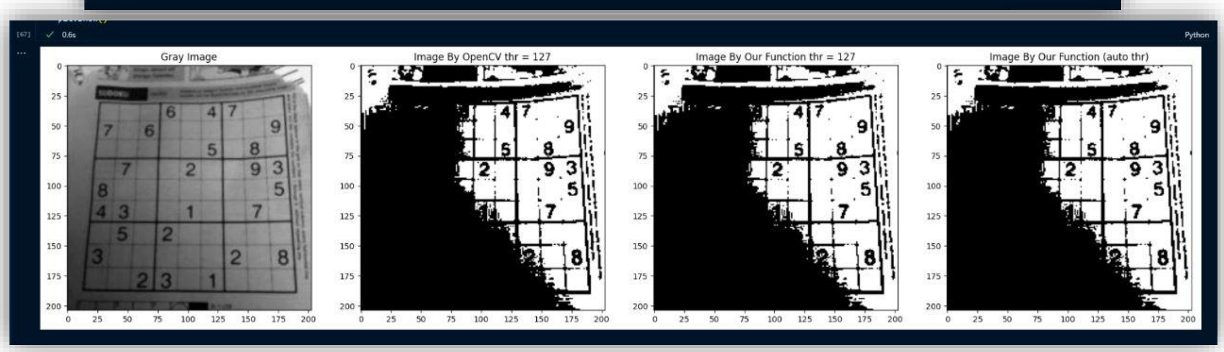
We have 2 functions for global thresholding: manual and automatic.

- **Manual:** it takes the threshold from the user which can be detected from the histogram of the image.
- **Automatic**: the threshold is calculated by getting the mean (average) of the pixels of the image.

For local threshold we divide the images into 9 parts and then calculate the mean (average) of each part and set the threshold for each part of them.

Generally for both types of thresholding pixels value above the threshold is set to max and for pixels value below the threshold is set to min.





Frequency domain filters:

Frequency filters process an image in the frequency domain. The image is Fourier transformed, multiplied with the filter function and then re-transformed into the spatial domain. Attenuating high frequencies results in a smoother image in the spatial domain, attenuating low frequencies enhances the edges.

We have two types of filter:

- **Low-pass filter :**

  It attenuates high frequencies and retains low frequencies unchanged.

- **High-pass filter :**

  It removes the low frequency components that means it keeps the high frequency components. It is used for sharpening the images.

Hybrid images:

To obtain a hybrid image from 2 images, we apply a high pass filter to one of them and low pass to the other and then we sum both of the results and get inverse fast Fourier transform.

Algorithm:

1- Get optimal DFT size as performance is best for image sizes that are multiple of the numbers two, three and five.

2- Convert our input image to float and expand it with another channel to hold the complex values:

3- Calculate DFT for image, rearrange the quadrants of the result, so that the origin (zero, zero).
4- Calculate the high or low pass filter of the image by looping over the image then choose the region which is needed to be filtered, then multiply the filter by the image.
5- Then we end up by the summation of the low pass-filtered image with the high pass - filtered image and get the fast Fourier transfom.