Cairo University

Faculty of Engineering

Systems and Biomedical Engineering

Computer Vision

Task 1 Report

Submitted to:

Dr. Ahmed M. Badawi

TA/ Eng. Laila Abbas.

TA/ Eng. Peter Salah.

Prepared by: Team 8.

Functions Implemented

1. *<u>Edge Detection:</u>*
    - Prewitt:
      The function takes the gray image after being blurred as input. It then convolves it with the two kernels in x and y direction for each pixel. Then sum the two arrays resulted to get array of gradient in x and y direction.

    - Robert:
      The function takes the gray image as input, divides it by 255 for scaling. It then convolves it with the two kernels in x and y direction for each pixel. Then sum the square root of the two arrays resulted to get array of gradient in x and y direction.

- Sobel:
  The function takes the gray image after being blurred as input. It then convolves it with the two kernels in x and y direction for each pixel using cv2 library. Then sum the two arrays resulted to get array of gradient in x and y direction.

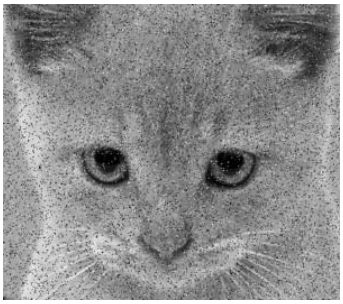
- Canny:
  The function takes the gray image after being blurred as input, applies Sobel operator in x and y using cv2 library. It then convolves it with the kernel for each pixel and setting the threshold from 100 to 200.

2. Add Noise:

- Uniform Noise:
  The function take image matrix & make another matrix of zeroes then apply a uniform noise (which is a random variable between two variable a, b then add theses noise to the original image matrix.

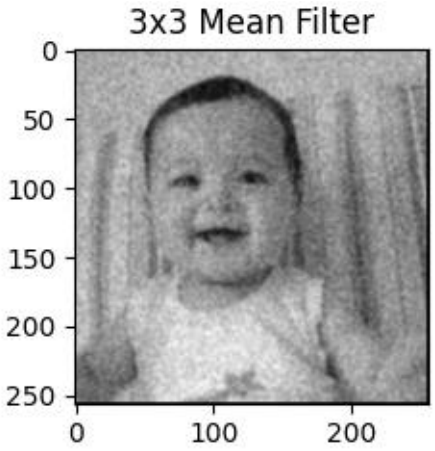
- Gaussian
  The function take mean and variance then calculate sigma form variance, we make another matrix which shape is the same as the original matrix and apply normal to it with value of mean and sigma ,then add this matrix to the original matrix.

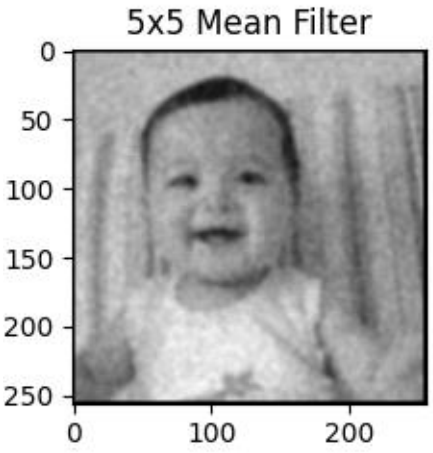
- salt & pepper
  In this function we choose a random number of pixels that we want to color white and the number of pixels we want to color black.



3. Filter Noise:

- Mean filter :
  The function takes the grayscale image and size of the filter as input. Then loop through the image pixels and by the number of pixels equal to the filter size the function summate these pixels and divide them by the filter size to get mean and put the mean pixel in the new array





- Median filter
  The function takes the grayscale image and size of the mask as input. Then set border for image in order to keep the image size then loop through the image pixels and by the number of pixels then sort these pixels to get median one to put it in new array

- Gaussian filter

  The function takes the grayscale image, size of kernel, and sigma as input.

  Then loop through the image pixels and by the number of pixels calculate the Gaussian equation and get the value of the image array

4. Equalization:

   In this function we calculate frequency count of each pixel then apply cumulative sum on it then apply normalization and reshape the matrix as the original image matrix then calculate the together.

5. Normalization:

The function takes the grayscale image as an input.

It normalizes each pixel in the image with the function:

(Pixel value – min_pixel) / (max_pixel- min_pixel)*255

6. Global & local Thresholding:

a. Global Thresholding :

The function takes the grayscale image and threshold as an input.

It replaces each pixel in the image with a black pixel if the pixel intensity is less than the threshold or a white pixel if the pixel intensity is greater than that threshold.

b. local Thresholding :

The function takes the grayscale image and 4 thresholds as an input.

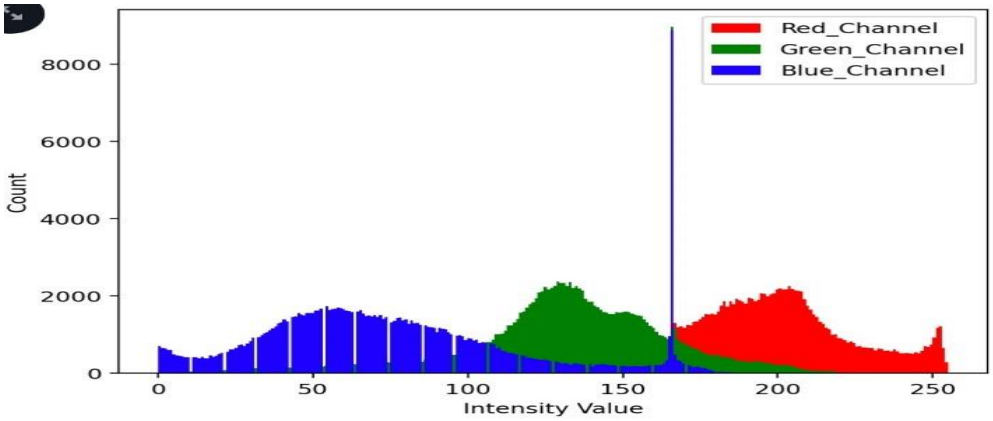
It divides the image into 4 parts then it applies the concept of

Thresholding for each part, then it concatenates the 4 parts.

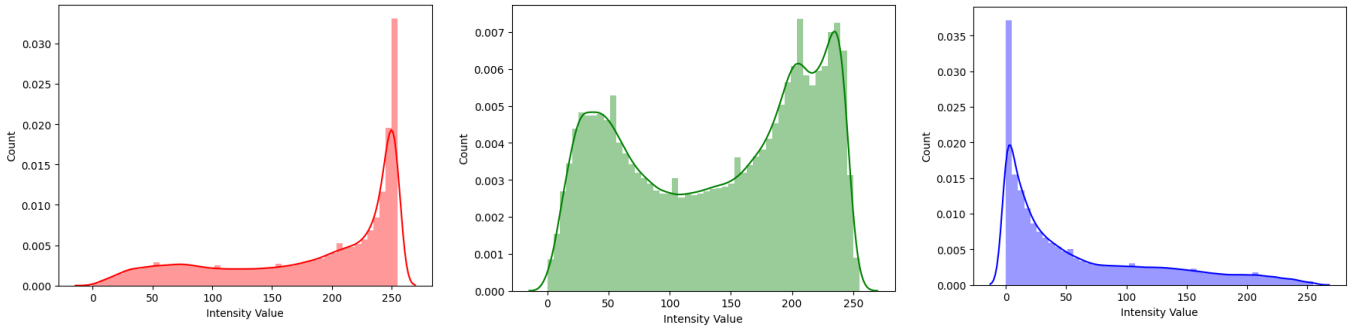
7. RGB Histogram:

- RGB Histogram:

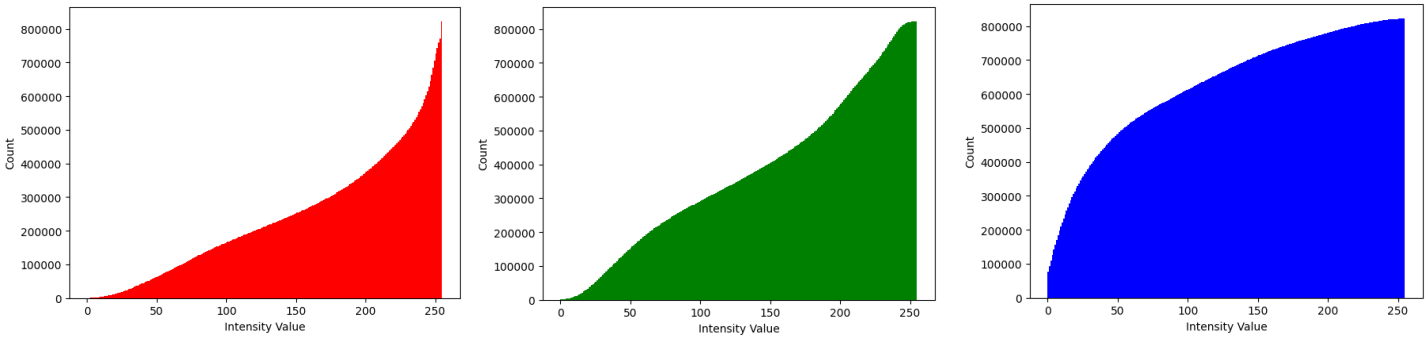
  The function takes a color image then split the three channel of the image and make an image array for everyone for the pixels values count their intensity and represent the intensity of every channel by using the histogram function



- Distribution function:
  The function takes a color image then split the three channel of the image and makes an image array for every color channel count by the pixels' intensity and represent by using displot function



- Cumulative function:
  The function takes a color image then split the three channel of the image and makes an image array for every color channel count by the pixels' cumulative intensity and represents by using hist function



8. Low & High Pass Filters:

a. Low Pass:

The Function takes the grayscale image as an input.

It converts the image from spatial domain to frequency domain by perform Fourier transform on the image.

Then generates the ideal low pass filter by using cutoff frequency.

And finally performs inverse Fourier transform.

b. High Pass:

The Function also takes the grayscale image as an input.

The ideal high pass filter is generated by subtracting the ideal low pass filter from 1.

And then performs inverse Fourier transform

9. Hybrid Image:

Hybrid image is generated by adding the image resulted from low pass filter with the one resulted from the high pass filter.