Batch: B1 **Roll No : 16010121045**

Experiment No: 8

Title: Apply neighbourhood processing techniques**:** low pass, high pass and median filtering in spatial domain on a digital image

Objective: To learn and understand the effects of filtering in spatial and frequency domain on images using Matlab.

Expected Outcome of Experiment:

CO	Outcome
CO4	Design & implement algorithms for digital image enhancement, segmentation & restoration.

Books/ Journals/ Websites referred:

1. http://www.mathworks.com/support/
2. www.math.mtu.edu/~msgocken/intro/intro.html.
3. R. C.Gonsales R.E.Woods, “Digital Image Processing”, Second edition, Pearson Education
4. S.Jayaraman, S Esakkirajan, T Veerakumar “Digital Image Processing “Mc Graw Hill.
5. S.Sridhar,”Digital Image processing”, oxford university press, 1st edition."

Filtering in Spatial Domain:

Low pass filtering as the name suggests removes the high frequency content from the image. It is used to remove noise present in the image. Mask for the low pass filter is:

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

One important thing to note from the spatial response is that all the coefficients are positive. We could also use 5 x 5 or 7 x 7 mask as per our requirement. We place a 3 x 3 mask on the image. We start from the left hand top corner. We cannot work with the borders and hence are normally left as they are. We then multiply each component of the image with the corresponding value of the mask. Add these values to get the response. Replace the centre pixel of the o/p image with these responses. We now shift the mask towards the right till we reach the end of the line and then move it downwards.

High pass filtering as the name suggests removes the low frequency content from the image. It is used to highlight fine detail in an image or to enhance detail that has been blurred. Mask for the high pass filter is:

-1/9	-1/9	-1/9
-1/9	8/9	-1/9
-1/9	-1/9	-1/9

One important thing to note from the spatial response is that sum of all the coefficients is zero. We could also use 5 x 5 or 7 x 7 mask as per our requirement. We place a 3 x 3 mask on the image. We start from the left hand top corner. We cannot work with the borders and hence are normally left as they are. We then multiply each component of the image with the corresponding value of the mask. Add these values to get the response. Replace the centre pixel of the o/p image with these responses. We now shift the mask towards the right till we reach the end of the line and then move it downwards.

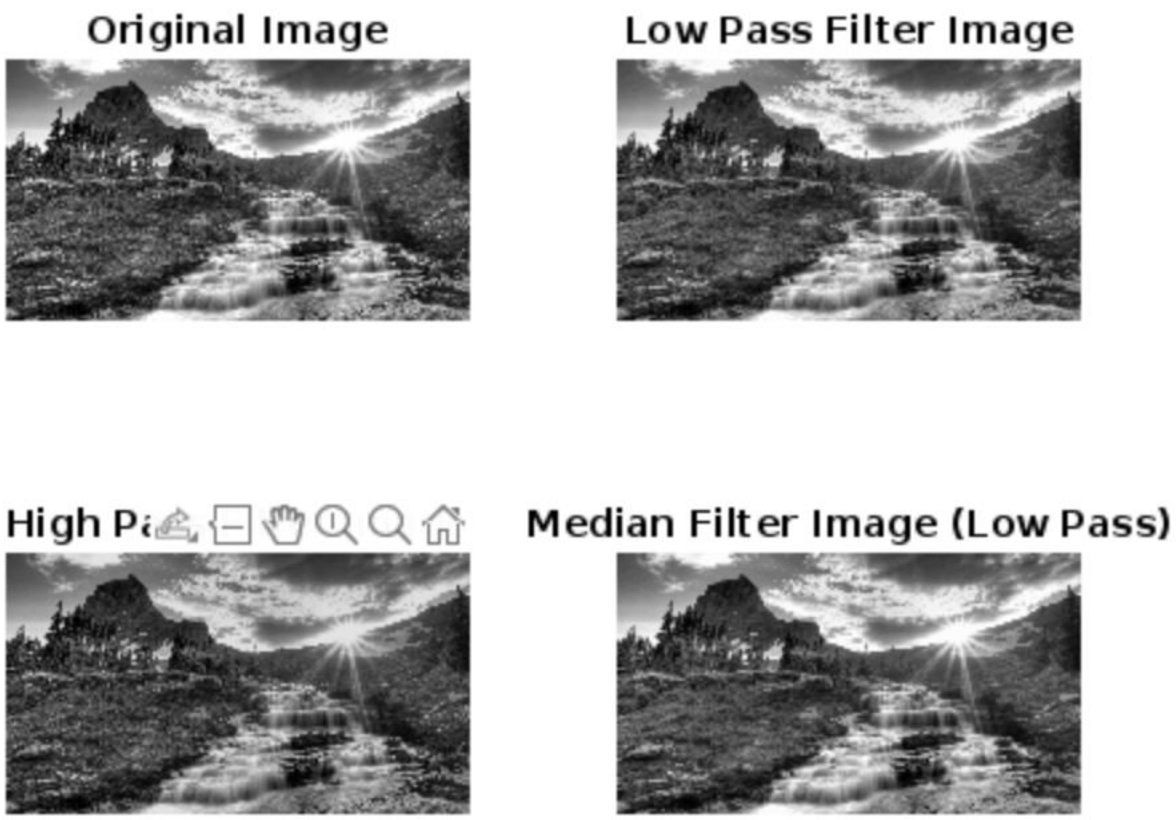
Median filtering is a signal processing technique developed by tukey that is useful for noise suppression in images. Here the input pixel is replaced by the median of the pixels contained in the window around the pixel. The median filter disregards extreme values and does not allow them to influence the selection of a pixel value which is truly representative of the neighbourhood.

Implementation Details:

Write Algorithm and Matlab commands used:

```
% Read Image
input_real = imread('example.jpg');
input = rgb2gray(input_real);

% Add zero padding
input = padarray(input, [1,1], 0, 'both');
[rows, columns] = size(input);

% Create low pass filter
lowpass = [1/9 1/9 1/9; 1/9 1/9 1/9; 1/9 1/9 1/9];
result_img_lowpass = uint8(zeros(rows, columns));

% Perform low pass filtering
for i = 2:rows-1
  for j = 2:columns-1
    new_matrix = [input(i-1,j-1) input(i-1,j) input(i-1,j+1); input(i,j-1) input(i,j) input(i,j+1); input(i+1,j-1) input(i+1,j) input(i+1,j+1)];
    temp_sum = 0;
    for k = 1:3
      for l = 1:3
        temp_sum = temp_sum + lowpass(k,l) * new_matrix(k,l);
      end
    end
    result_img_lowpass(i,j) = temp_sum;
  end
end

% Create high pass filter
highpass = [-1/9 -1/9 -1/9; -1/9 8/9 -1/9; -1/9 -1/9 -1/9];
result_img_highpass = uint8(zeros(rows, columns));

% Perform high pass filtering
for i = 2:rows-1
  for j = 2:columns-1
    new_matrix = [input(i-1,j-1) input(i-1,j) input(i-1,j+1); input(i,j-1) input(i,j) input(i,j+1); input(i+1,j-1) input(i+1,j) input(i+1,j+1)];
    temp_sum = 0;
    for k = 1:3
```

```
            for l = 1:3
                temp_sum = temp_sum + highpass(k,l) * new_matrix(k,l);
            end
        end
        if temp_sum < 0
            temp_sum = 0;
        end
        result_img_highpass(i,j) = temp_sum;
    end
end

% Apply median filter to low pass and high pass filtered images
result_img_lowpass_median = medfilt2(result_img_lowpass);
result_img_highpass_median = medfilt2(result_img_highpass);

% Display original and filtered images
subplot(2,2,1), imshow(input); title('Original Image');
subplot(2,2,2), imshow(result_img_lowpass); title('Low Pass Filter Image');
subplot(2,2,3), imshow(result_img_highpass); title('High Pass Filter Image');
subplot(2,2,4), imshow(result_img_lowpass_median); title('Median Filter Image (Low Pass)');
```



Conclusion:- Through this experiment we learnt the implementation of filters.

Post Lab Descriptive Questions

1. **List & explain different types of noise associated with a digital signal.**

Gaussian Noise: Also known as white noise, Gaussian noise follows a Gaussian distribution and has constant power spectral density across all frequencies. It arises from random variations in the signal due to factors such as electronic components, thermal effects, and atmospheric disturbances.
Impulse Noise: Consists of sudden, sharp spikes or impulses in the signal amplitude, often resulting from external interference, faulty equipment, or transmission errors. Impulse noise can corrupt individual signal samples and cause errors in signal processing.
Periodic Noise: Repeats at regular intervals and can be caused by electrical interference from power lines, machinery, or other sources. It manifests as unwanted periodic variations in the signal amplitude or frequency.
Shot Noise: Arises from the random arrival of discrete particles or photons in electronic devices, such as photodetectors or semiconductor devices. It introducesfluctuations in signal intensity, especially in low-light imaging systems.
Quantization Noise: Occurs in digital systems during analog-to-digital conversion due to the finite precision of the conversion process, leading to rounding errors or discrepancies between the original analog signal and its quantized representation.

2. **Explain with the help of an example how filtering helps in enhancing the quality of an image.**

Filtering enhances image quality by mitigating noise, emphasizing features, and improving overall clarity. For instance, consider an image of a landscape with added Gaussian noise. Filtering techniques like Gaussian smoothing or median filtering can effectively reduce noise, resulting in a cleaner and visually appealing image. Additionally, filters such as Sobel or Prewitt edge detection enhance edges, while Laplacian of Gaussian filters extract specific features like textures or corners, making them more prominent and easier to analyze. Moreover, smoothing filters like Gaussian blur can achieve a pleasing aesthetic by reducing minor imperfections. Ultimately, filtering plays a vital role in enhancing image quality across various applications by refining visual details and improving interpretability.