A Comparative Study of various Noise Removal Techniques

Group Members:

KARTIK GOEL-19BCE2002

VANSH AGGARWAL-19BCE0987

TANISHQ SHAH-19BCE2067

Faculty Name:

PROF.AKILA VICTOR

Assistant Professor, SCOPE, VIT Vellore

FALL-SEMESTER

JULY 2020-NOVEMBER 2020

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Table of Contents

Synopsis

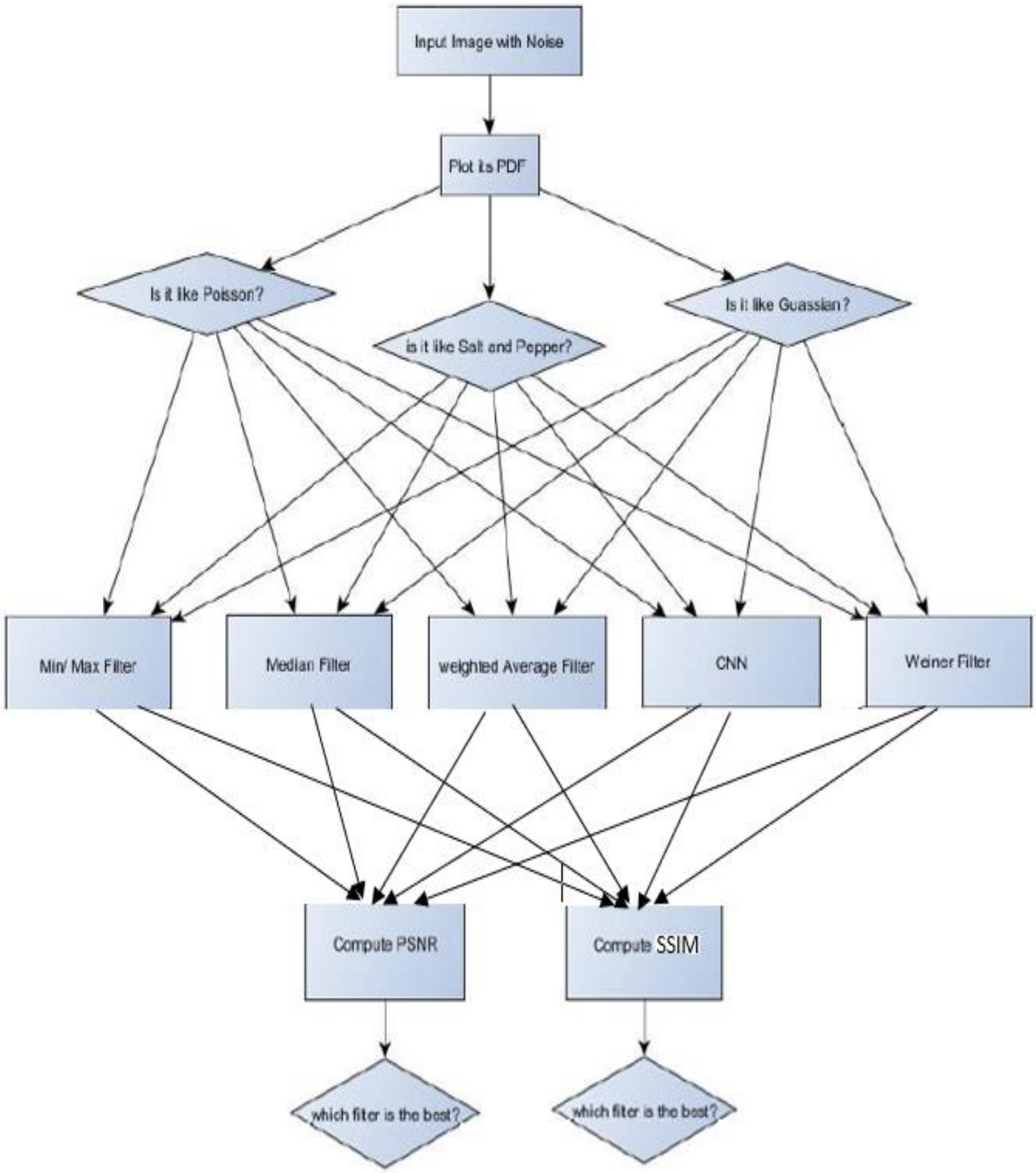
Noise is an unwanted signal. In an image noise is a random variation of brightness or color information and is usually an aspect of electronic noise. It can be produced by the image sensor and circuitry of a scanner or digital camera. Image noise can also originate in film grain and in the unavoidable shot noise of an ideal photon detector. Image noise is an undesirable byproduct of image capture that obscures the desired information. Image noise can range from almost imperceptible specks on a digital photograph taken in good light, to optical and radio astronomical images that are almost entirely noise, from which a small amount of information can be derived by sophisticated processing. Such a noise level would be unacceptable in a photograph since it would be impossible even to determine the subject. Hence we need Noise removal techniques to get clear pictures. Noise reduction is the process of removing noise from a signal. Noise reduction techniques exist for audio and images. Noise reduction algorithms tend to alter signals to a greater or lesser degree.

Objective

This project aims to determine ways through which one can detect the type of noise present in the image. In order to identify the same, we have taken sample images with noise and plotted their Probability Density Functions (PDF) to analyse the shape of PDFs with respect to the type of noise present in the picture. On comparing these graphs one can see that a particular type of noise has characteristic PDFs which can help detect the type of noise present in an

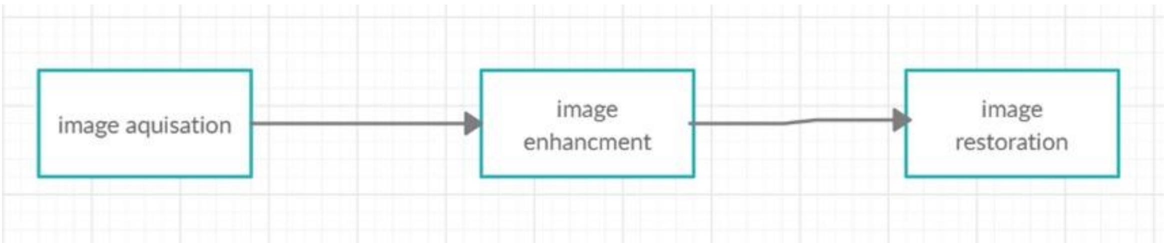
image. Further, we have applied filters such as Min/ Max Filter, Median Filter, Weiner Filter, Convolutional Neural Network Filter, Weighted Average Filter on each of these noises. Peak- Signal-to-Noise-Ratio (PSNR) was plotted for each filter applied to the noise image to compare which filter is the best to remove a particular type of noise present in the image. The filter having the highest PSNR value for a noisy image will be the best filter to use. The results obtained are finally analyzed in the project.

Abstract



Input Image with Noise
Plot its PDF
Is it like Poisson?
is it like Salt and Pepper?
Is it like Guassian?
Min/ Max Filter
Median Filter
weighted Average Filter
CNN
Weiner Filter
Compute PSNR
Compute SSIM
which filter is the best?
which filter is the best?

BLOCK-DIAGRAM



INTRODUCTION

We see various images in our day to day life. Those images might be blur and unclear or perfectly clear but that of course is ideal. There are three types of noises that we have used in our project.

1) **Salt and pepper-** Salt-and-pepper noise is a form of noise sometimes seen on images. It is also known as impulse noise. This noise can be caused by sharp and sudden disturbances in the image signal. It presents itself as sparsely occurring white and black pixels. An effective noise reduction method for this type of noise is a median filter or a morphological filter.

2) Gaussian- Gaussian noise is statistical noise having a probability density function (PDF) equal to that of the normal distribution, which is also known as the Gaussian distribution. In other words, the values that the noise can take on are Gaussian-distributed.

3) Poisson- Poisson or shot photon noise is the noise that can cause, when number of photons sensed by the sensor is not sufficient to provide detectable statistical information. This noise has root mean square value proportional to square root intensity of the image. Different pixels are suffered by independent noise values. At practical grounds the photon noise and other sensor based noise corrupt the signal at different proportions.

Technique used for Identification of the type of noise:

A probability Density function (PDF) is characteristic for determining the type of noise present in an image. Under this project, we will be plotting the PDFs of the images containing noise and try to identify their characteristic functions. When a random image is taken, the PDF for the same will be plotted and its shape will be analyzed to see which types of noise is present in the image.

Techniques used for noise removal:

1. Filtering Techniques

Filtering is a technique for modifying or enhancing an image. For example, you can filter an image to emphasize certain features or remove other features.

Types of filters used:

>Linear filter - Linear filtering is filtering in which the value of an output pixel is a linear combination of the values of the pixels in the input pixel's neighborhood. In linear filtering we use weighted average filter. In weighted average filter, we gave more weight to the center value, due to which the contribution of center becomes more than the rest of the values.

> Weighted average filter: In linear filtering we use weighted average filter. In weighted average filter, we gave more weight to the center value, due to which the contribution of center becomes more than the rest of the values. Due to weighted average filtering, we can control the blurring of image.

>Non-linear filter –Nonlinear filters are not employed in image processing as frequently as linear filters, because in many cases they are based on heuristics closely tied to the particular application. The general idea in non-linear image filtering is that instead of using the spatial mask in a convolution process, the mask is used to obtain the neighbouring pixel values, and then ordering mechanisms produce the output pixel.

> **Median filter**: Median filtering is a nonlinear method used to remove noise from images. It is widely used as it is very effective at removing noise while preserving edges. It is particularly effective at removing ‘salt and pepper’ type noise. The median filter works by moving through the image pixel by pixel, replacing each value with the median value of neighbouring pixels.

> Max and min filter: Maximum and minimum filters attribute to each pixel in an image a new value equal to the maximum or minimum value in a neighborhood around that pixel. The neighbourhood stands for the shape of the filter. Maximum and minimum filters have been used in contrast enhancement and normalization.

>Wiener filter: The most important technique for removal of blur in images due to linear motion or unfocussed optics is the Wiener filter. From a signal processing standpoint, blurring due to linear motion in a photograph is the result of poor sampling. Each pixel in a digital representation of the photograph should represent the intensity of a single stationary point in front of the camera.

Unfortunately, if the shutter speed is too slow and the camera is in motion, a given pixel will be an amalgam of intensities from points along the line of the camera's motion. This is a two-dimensional analogy to

$$G(u,v)=F(u,v).H(u,v)$$

where F is the Fourier transform of an "ideal" version of a given image, and H is the blurring function. In this case H is a sine function: if three pixels in a line contain info from the same point on an image, the digital image will seem to have been convolved with a three-point boxcar in the time domain. Ideally one could reverse-engineer a Fest, or F estimate, if G and H are known. This technique is inverse filtering.

2.Convolutional Neural Network with Back Propagation Algorithm:

In deep learning, a convolutional neural network is a class of deep, feed-forward artificial neural networks, most commonly applied to analysing visual imagery. CNNs use a variation of multilayer perceptron designed to require minimal pre-processing. They are also known as shift invariant or space invariant artificial neural networks (SIANN), based on their shared weights architecture and invariance characteristics.

Techniques used for Measuring the accuracy of Image Noise Removal:

- **PSNR**: Peak signal-to-noise ratio, often abbreviated PSNR, is an engineering term for the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. Higher the PSNR value of a filtered image, better that filter is in removing the noise present in the image.

- **SSIM**: The structural similarity index measure (SSIM) is a method for predicting the perceived quality of digital television and cinematic pictures, as well as other kinds of digital images and videos. SSIM is used for measuring the similarity between two images. The SSIM index is a full reference metric; in other words, the measurement or prediction of image quality is based on an initial uncompressed or distortion-free image as reference.

Our project will contain the following:

- Sample input images without noises
- Probability Density Function (PDF) graphs of the images with Noise
- Sample input images with Noise
- Code for Min/ Max Filter, Median Filter, Convolutional Neural Network (CNN) Filter, Weighted Average Filter, Weiner Filter
- PSNR and SSIM values after applying the above mentioned filters to the images with noises.

CODE

```
clc
clear all;
% image_folder='C:\Users\karti\Desktop\ip_project';
image_folder='C:\Users\Vansh Aggarwal\Desktop\image processing project';
filenames=dir(fullfile(image_folder,'*.jpg'))
total_images=numel(filenames);
n=1;
a=1;
flag=1;
```

```
for i=1:total_images
 add=5+11*(i-1);
 col_1=strcat("C",num2str(add));
  col_2=strcat("D",num2str(add));
   col_3=strcat("E",num2str(add));
    col_4=strcat("F",num2str(add));
     col_5=strcat("G",num2str(add));
      col_6=strcat("H",num2str(add));
       col_7=strcat("I",num2str(add));
    xlswrite('image.xlsx',{'''NOISES'},'Sheet1',col_1);
       xlswrite('image.xlsx',{'''SnP Noise'},'Sheet1',col_2);
          xlswrite('image.xlsx',{'''Gauss Noises'},'Sheet1',col_3);
          xlswrite('image.xlsx',{'''Poission Noises'},'Sheet1',col_4);
            xlswrite('image.xlsx',{'''SnP Noise'},'Sheet1',col_5);
          xlswrite('image.xlsx',{'''Gauss Noises'},'Sheet1',col_6);
          xlswrite('image.xlsx',{'''Poission Noises'},'Sheet1',col_7);
end



ch='y';
while(ch=='y')
    flt = input('enter which filter do u wanna see: ', 's');
      if(strcmp(flt,'wiener')==1)
          for i=1:total_images
            f=fullfile(image_folder, filenames(i).name);
            img_orig=imread(f);
            c_img=img_orig;
            img_orig=rgb2gray(img_orig);
            img_snp=imnoise(img_orig,'salt & pepper' ,0.02)
            img_gauss=imnoise(img_orig,'gaussian',0,0.007)
            img_pos=imnoise(img_orig,'poisson')
            L1=wiener2(img_gauss,[5 5]);
            L2=wiener2(img_pos,[5 5]);
            L3=wiener2(img_snp,[5 5]);
         figure;

     subplot(1,7,1);imshow(c_img);title('Original image')
    subplot(1,7,2);imshow(img_gauss);title('gausian Noisy image')
    subplot(1,7,3);imshow(L1);title('mins_Denoised Gausian image')
    subplot(1,7,4);imshow(img_pos);title('Possion Noisy image')
    subplot(1,7,5);imshow(L2);title('mins_Denoised Poison image')
    subplot(1,7,6);imshow(img_snp);title('snp Noisy image')
    subplot(1,7,7);imshow(L3);title('mins_Denoised snp image')

        noisyGPSNR(i)=psnr(L1,img_orig);
         ssim_noisyGPSNR(i)=ssim(L1,img_orig);
          noisyPPSNR(i)=psnr(L2,img_orig);
          ssim_noisyPPSNR(i)=ssim(L2,img_orig);
         noisySPSNR(i)=psnr(L3,img_orig);
          ssim_noisySPSNR(i)=ssim(L3,img_orig);
          end

          for j=1:total_images
              disp("For image:"+j);
              disp("Gaussian Noisy Image PSNR"+noisyGPSNR(j));
```

```
        disp("Poission Noisy Image PSNR"+noisyPPSNR(j));
        disp("Snp Noisy Image PSNR"+noisySPSNR(j));
      end

    for i=1:total_images
        add=6+11*(i-1);
        row_heading=strcat("C",num2str(add));
        snp_ad=strcat("D",num2str(add));
         gaus_ad=strcat("E",num2str(add));
         pos_ad=strcat("F",num2str(add));
%           snr
         ssim_snp_ad=strcat("G",num2str(add));
         ssim_gaus_ad=strcat("H",num2str(add));
         ssim_pos_ad=strcat("I",num2str(add));
         xlswrite('image.xlsx',{'wiener Filter'},'Sheet1',row_heading);
       xlswrite('image.xlsx',noisySPSNR(i),'Sheet1',snp_ad);
      xlswrite('image.xlsx',noisyGPSNR(i),'Sheet1',gaus_ad);
      xlswrite('image.xlsx',noisyPPSNR(i),'Sheet1',pos_ad);
       xlswrite('image.xlsx',ssim_noisySPSNR(i),'Sheet1',ssim_snp_ad);
      xlswrite('image.xlsx',ssim_noisyGPSNR(i),'Sheet1',ssim_gaus_ad);
      xlswrite('image.xlsx',ssim_noisyPPSNR(i),'Sheet1',ssim_pos_ad);
    end
   elseif(strcmp(flt,'min')==1)
       for k=1:total_images
          f=fullfile(image_folder, filenames(k).name);
          img_orig=imread(f);
          c_img=img_orig
          img_orig=im2double(img_orig);
          img_orig=rgb2gray(img_orig);
          img_snp=imnoise(img_orig,'salt & pepper' ,0.02)
          img_gauss=imnoise(img_orig,'gaussian',0,0.005)
          img_pos=imnoise(img_orig,'poisson')
          mins1=mins(img_gauss);
          mins2=mins(img_pos);
          mins3=mins(img_snp);

      figure;

  subplot(1,7,1);imshow(c_img);title('Original image')
 subplot(1,7,2);imshow(img_gauss);title('gausian Noisy image')
 subplot(1,7,3);imshow(mins1);title('mins Denoised Gausian image')
 subplot(1,7,4);imshow(img_pos);title('Possion Noisy image')
 subplot(1,7,5);imshow(mins2);title('mins Denoised Poison image')
 subplot(1,7,6);imshow(img_snp);title('snp Noisy image')
 subplot(1,7,7);imshow(mins3);title('mins Denoised snp image')
            min_noisySPSNR(k)=psnr(mins3,img_orig);
            ssim_min_noisySPSNR(k)=ssim(mins3,img_orig);
             min_noisyGPSNR(k)=psnr(mins1,img_orig);
             ssim_min_noisyGPSNR(k)=ssim(mins1,img_orig);
                min_noisyPPSNR(k)=psnr(mins2,img_orig);
                 ssim_min_noisyPPSNR(k)=ssim(mins2,img_orig);
       end
          for j=1:total_images
         disp("For image:"+j);
         disp("Min filter Gaussian Noisy Image PSNR"+min_noisyGPSNR(j));
         disp(" Min Poission Noisy Image PSNR"+min_noisyPPSNR(j));
```

```
          disp(" Min Snp Noisy Image PSNR"+min_noisySPSNR(j));
           end
         for i=1:total_images
         add=7+11*(i-1);
         row_heading=strcat("C",num2str(add));
         snp_ad=strcat("D",num2str(add));
          gaus_ad=strcat("E",num2str(add));
          pos_ad=strcat("F",num2str(add));
%            ssim
          ssim_snp_ad=strcat("G",num2str(add));
          ssim_gaus_ad=strcat("H",num2str(add));
          ssim_pos_ad=strcat("I",num2str(add));
          xlswrite('image.xlsx',{'Min Filter'},'Sheet1',row_heading);
        xlswrite('image.xlsx',min_noisySPSNR(i),'Sheet1',snp_ad);
       xlswrite('image.xlsx',min_noisyGPSNR(i),'Sheet1',gaus_ad);
       xlswrite('image.xlsx',min_noisyPPSNR(i),'Sheet1',pos_ad);
        xlswrite('image.xlsx',ssim_min_noisySPSNR(i),'Sheet1',ssim_snp_ad);
       xlswrite('image.xlsx',ssim_min_noisyGPSNR(i),'Sheet1',ssim_gaus_ad);
       xlswrite('image.xlsx',ssim_min_noisyPPSNR(i),'Sheet1',ssim_pos_ad);
         end
       elseif(strcmp(flt,'max')==1)
        for k=1:total_images
            f=fullfile(image_folder, filenames(k).name);
            img_orig=imread(f);
             c_img=img_orig;
            img_orig=im2double(img_orig);
            img_orig=rgb2gray(img_orig);
            img_snp=imnoise(img_orig,'salt & pepper' ,0.02)
            img_gauss=imnoise(img_orig,'gaussian',0,0.005)
            img_pos=imnoise(img_orig,'poisson')
            maxs1=maxs(img_gauss);
            maxs2=maxs(img_pos);
            maxs3=maxs(img_snp);
       figure;

  subplot(1,7,1);imshow(c_img);title('Original image')
 subplot(1,7,2);imshow(img_gauss);title('gausian Noisy image')
 subplot(1,7,3);imshow(maxs1);title('max Denoised Gausian image')
 subplot(1,7,4);imshow(img_pos);title('Possion Noisy image')
 subplot(1,7,5);imshow(maxs2);title('max Denoised Poison image')
 subplot(1,7,6);imshow(img_snp);title('snp Noisy image')
 subplot(1,7,7);imshow(maxs3);title('max Denoised snp image')
            max_noisySPSNR(k)=psnr(maxs3,img_orig);
            ssim_max_noisySPSNR(k)=ssim(maxs3,img_orig);
            max_noisyGPSNR(k)=psnr(maxs1,img_orig);
             ssim_max_noisyGPSNR(k)=ssim(maxs1,img_orig);
            max_noisyPPSNR(k)=psnr(maxs2,img_orig);
            ssim_max_noisyPPSNR(k)=ssim(maxs2,img_orig);
        end

       for j=1:total_images
          disp("For image:"+j);
          disp("Max filter Gaussian Noisy Image PSNR"+max_noisyGPSNR(j));
          disp(" Max Poission Noisy Image PSNR"+max_noisyPPSNR(j));
          disp(" Max Snp Noisy Image PSNR"+max_noisySPSNR(j));
       end
```

```
        for i=1:total_images
         add=8+11*(i-1);
         row_heading=strcat("C",num2str(add));
         snp_ad=strcat("D",num2str(add));
          gaus_ad=strcat("E",num2str(add));
          pos_ad=strcat("F",num2str(add));
%            ssim
          ssim_snp_ad=strcat("G",num2str(add));
          ssim_gaus_ad=strcat("H",num2str(add));
          ssim_pos_ad=strcat("I",num2str(add));
           xlswrite('image.xlsx',{'Max Filter'},'Sheet1',row_heading);
         xlswrite('image.xlsx',max_noisySPSNR(i),'Sheet1',snp_ad);
        xlswrite('image.xlsx',max_noisyGPSNR(i),'Sheet1',gaus_ad);
        xlswrite('image.xlsx',max_noisyPPSNR(i),'Sheet1',pos_ad);

         xlswrite('image.xlsx',ssim_max_noisySPSNR(i),'Sheet1',ssim_snp_ad);
        xlswrite('image.xlsx',ssim_max_noisyGPSNR(i),'Sheet1',ssim_gaus_ad);
        xlswrite('image.xlsx',ssim_max_noisyPPSNR(i),'Sheet1',ssim_pos_ad);
        end
    elseif(strcmp(flt,'waf')==1)
         for i=1:total_images
            f=fullfile(image_folder, filenames(i).name);
            img_orig=imread(f);
            img_orig=rgb2gray(img_orig);
            img_snp=imnoise(img_orig,'salt & pepper' ,0.02)
            img_gauss=imnoise(img_orig,'gaussian',0,0.005)
            img_pos=imnoise(img_orig,'poisson')
            af1=waf(img_gauss);
            af2=waf(img_pos);
            af3=waf(img_snp);
            waf_noisyGPSNR(i)=psnr(af1,img_orig);
            ssim_waf_noisyGPSNR(i)=ssim(af1,img_orig);
            waf_noisyPPSNR(i)=psnr(af2,img_orig);
            ssim_waf_noisyPPSNR(i)=ssim(af2,img_orig);
            waf_noisySPSNR(i)=psnr(af3,img_orig);
            ssim_waf_noisySPSNR(i)=ssim(af3,img_orig);
figure;
    subplot(1,7,1);imshow(f);title('Original image')
    subplot(1,7,2);imshow(img_gauss);title(' gausian Noisy image')
    subplot(1,7,3);imshow(uint8(af1));title('Denoised Gausian image')
    subplot(1,7,4);imshow(img_pos);title('Possion Noisy image')
    subplot(1,7,5);imshow(uint8(af2));title('Denoised Poison image')
    subplot(1,7,6);imshow(img_snp);title('snp Noisy image')
    subplot(1,7,7);imshow(uint8(af3));title('Denoised snp image')
         end

         for j=1:total_images
           disp("For image:"+j);
           disp("Weighted filter Gaussian Noisy Image PSNR"+waf_noisyGPSNR(j));
           disp("Weighted Poission Noisy Image PSNR"+waf_noisyPPSNR(j));
           disp("Weighted Min Snp Noisy Image PSNR"+waf_noisySPSNR(j));
         end
        for i=1:total_images
         add=9+11*(i-1);
```

```
         row_heading=strcat("C", num2str(add));
         snp_ad=strcat("D",num2str(add));
          gaus_ad=strcat("E",num2str(add));
          pos_ad=strcat("F",num2str(add));
%            ssim
          ssim_snp_ad=strcat("G",num2str(add));
          ssim_gaus_ad=strcat("H",num2str(add));
          ssim_pos_ad=strcat("I",num2str(add));
       xlswrite('image.xlsx',{'WAF'},'Sheet1',row_heading);
        xlswrite('image.xlsx',waf_noisySPSNR(i),'Sheet1',snp_ad);
        xlswrite('image.xlsx',waf_noisyGPSNR(i),'Sheet1',gaus_ad);
        xlswrite('image.xlsx',waf_noisyPPSNR(i),'Sheet1',pos_ad);
        xlswrite('image.xlsx',ssim_waf_noisySPSNR(i),'Sheet1',ssim_snp_ad);
        xlswrite('image.xlsx',ssim_waf_noisyGPSNR(i),'Sheet1',ssim_gaus_ad);
        xlswrite('image.xlsx',ssim_waf_noisyPPSNR(i),'Sheet1',ssim_pos_ad);
        end
     elseif(strcmp(flt,'cnn')==1)
        for i=1:total_images
         f=fullfile(image_folder, filenames(i).name);
         pristineRGB=imread(f);
         pristineRGB=im2double(pristineRGB);
img_snp=imnoise(pristineRGB,'Salt & Pepper',0.07);
img_gauss=imnoise(pristineRGB,'gaussian',0,0.005)
img_pos=imnoise(pristineRGB,'poisson')
cnn1=cnn(img_snp);
cnn2=cnn(img_gauss);
cnn3=cnn(img_pos);
 cnn_noisyGPSNR(i)=psnr(cnn2,pristineRGB);
 ssim_cnn_noisyGPSNR(i)=ssim(cnn2,pristineRGB);
 cnn_noisyPPSNR(i)=psnr(cnn3,pristineRGB);
  ssim_cnn_noisyPPSNR(i)=ssim(cnn3,pristineRGB);
 cnn_noisySPSNR(i)=psnr(cnn1,pristineRGB);
 ssim_cnn_noisySPSNR(i)=ssim(cnn1,pristineRGB);
  figure;
   subplot(1,7,1);imshow(pristineRGB);title('Original image')
   subplot(1,7,2);imshow(img_gauss);title(' gausian Noisy image')
   subplot(1,7,3);imshow(cnn2);title('Denoised Gausian image')
   subplot(1,7,4);imshow(img_pos);title('Possion Noisy image')
   subplot(1,7,5);imshow(cnn3);title('Denoised Poison image')
   subplot(1,7,6);imshow(img_snp);title('snp Noisy image')
   subplot(1,7,7);imshow(cnn1);title('Denoised snp image')
        end

         for j=1:total_images
           disp("For image:"+j);
           disp("CNN filter Gaussian Noisy Image PSNR"+cnn_noisyGPSNR(j));
           disp("CNN Filter Poission Noisy Image PSNR"+cnn_noisyPPSNR(j));
           disp("CNN Filter Snp Noisy Image PSNR"+cnn_noisySPSNR(j));
         end

  for i=1:total_images
         add=10+11*(i-1)
         row_heading=strcat("C",num2str(add));
         snp_ad=strcat("D",num2str(add));
          gaus_ad=strcat("E",num2str(add));
          pos_ad=strcat("F",num2str(add));
```

```
%           ssim
          ssim_snp_ad=strcat("G",num2str(add));
          ssim_gaus_ad=strcat("H",num2str(add));
          ssim_pos_ad=strcat("I",num2str(add));
         xlswrite('image.xlsx',{'CNN'},'Sheet1',row_heading);
         xlswrite('image.xlsx',cnn_noisySPSNR(i),'Sheet1',snp_ad);
        xlswrite('image.xlsx',cnn_noisyGPSNR(i),'Sheet1',gaus_ad);
        xlswrite('image.xlsx',cnn_noisyPPSNR(i),'Sheet1',pos_ad);
        xlswrite('image.xlsx',ssim_cnn_noisySPSNR(i),'Sheet1',ssim_snp_ad);
        xlswrite('image.xlsx',ssim_cnn_noisyGPSNR(i),'Sheet1',ssim_gaus_ad);
        xlswrite('image.xlsx',ssim_cnn_noisyPPSNR(i),'Sheet1',ssim_pos_ad);
 end



       elseif(strcmp(flt,'median')==1)
        for i=1:total_images
         f=fullfile(image_folder, filenames(i).name);
         img_orig=imread(f);
         img_orig=rgb2gray(img_orig);
      img_snp=imnoise(img_orig,'Salt & Pepper',0.07);
  img_gauss=imnoise(img_orig,'gaussian',0,0.005)
  img_pos=imnoise(img_orig,'poisson')
  med1=median(img_snp);
  med2=median(img_gauss);
  med3=median(img_pos);
   med_noisyGPSNR(i)=psnr(med2,img_orig);
   med_ssimG(i)=ssim(med2,img_orig);
   med_noisyPPSNR(i)=psnr(med3,img_orig);
   med_ssimP(i)=ssim(med3,img_orig);
   med_noisySPSNR(i)=psnr(med1,img_orig);
   med_ssimS(i)=ssim(med1,img_orig);
     figure;
      subplot(1,7,1);imshow(img_orig);title('Original image')
      subplot(1,7,2);imshow(img_gauss);title(' gausian Noisy image')
      subplot(1,7,3);imshow(med2);title('Denoised Gausian image')
      subplot(1,7,4);imshow(img_pos);title('Possion Noisy image')
      subplot(1,7,5);imshow(med3);title('Denoised Poison image')
      subplot(1,7,6);imshow(img_snp);title('snp Noisy image')
      subplot(1,7,7);imshow(med1);title('Denoised snp image')
            end

          for j=1:total_images
            disp("For image:"+j);
            disp(" filter Gaussian Noisy Image PSNR"+med_noisyGPSNR(j));
            disp("CNN Filter Poission Noisy Image PSNR"+med_noisyPPSNR(j));
            disp("CNN Filter Snp Noisy Image PSNR"+med_noisySPSNR(j));
          end
         for i=1:total_images
          add=11+11*(i-1);
          row_heading=strcat("C",num2str(add));
          snp_ad=strcat("D",num2str(add));
           gaus_ad=strcat("E",num2str(add));
           pos_ad=strcat("F",num2str(add));
%           ssim
           ssim_snp_ad=strcat("G",num2str(add));
```

```
            ssim_gaus_ad=strcat("H",num2str(add));
            ssim_pos_ad=strcat("I",num2str(add));
           xlswrite('image.xlsx',{'median'},'Sheet1',row_heading);
          xlswrite('image.xlsx',med_noisySPSNR(i),'Sheet1',snp_ad);
         xlswrite('image.xlsx',med_noisyGPSNR(i),'Sheet1',gaus_ad);
         xlswrite('image.xlsx',med_noisyPPSNR(i),'Sheet1',pos_ad);
          xlswrite('image.xlsx',med_ssimP(i),'Sheet1',ssim_snp_ad);
         xlswrite('image.xlsx',med_ssimG(i),'Sheet1',ssim_gaus_ad);
         xlswrite('image.xlsx',med_ssimP(i),'Sheet1',ssim_pos_ad);

          end
      end
    ch=input('do u wanna use more images(y/n)', 's');
end


  for i=1:total_images
      f=fullfile(image_folder, filenames(i).name);
      img_orig=imread(f);
      img_orig=rgb2gray(img_orig);
P = imnoise(img_orig,'poisson');
snp=imnoise(img_orig,'salt & pepper' ,0.02)
G=imnoise(img_orig,'gaussian',0,0.005)

[r_pos c_pos]=size(P);
[counts_pos,binLocations_pos] = imhist(P);
pdf_pos=counts_pos/numel(P);

[r_snp c_snp]=size(snp);
[counts_snp,binLocations_snp] = imhist(snp);
pdf_snp=counts_snp/numel(snp);

[r_g c_g]=size(G);
[counts_g,binLocations_g] = imhist(G);
pdf_g=counts_g/numel(G);

figure
     subplot(1,4,1);plot(binLocations_pos, pdf_pos, 'b-', 'LineWidth', 2);xlabel('Gray Levels');
ylabel('Pixels Probability'); title('Poisson'); grid on;

 subplot(1,4,2);plot(binLocations_g, pdf_g, 'b-', 'LineWidth', 2);xlabel('Gray Levels');
ylabel('Pixels Probability'); title('Gaussian'); grid on;


 subplot(1,4,3);plot(binLocations_snp, pdf_snp, 'b-', 'LineWidth', 2);xlabel('Gray Levels');
ylabel('Pixels Probability'); title('SNP'); grid on;

subplot(1,4,4);imshow(img_orig); title('Image');

  end

function minss=mins(img_n)
        img_n=im2double(img_n);
        [r c]=size(img_n);
        ImL=zeros(r,c);
        ImU=zeros(r,c);
```

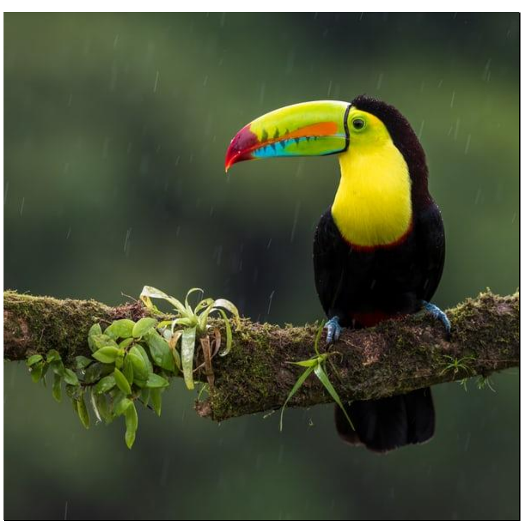
```
        for i=2:r-1
            for j=2:c-1
                M=img_n(i==1:i+1,j-1:j+1);
                Min=min(M(:));
                Max=max(M(:));
                ImL(i,j)=Min;
%                ImU(i,j)=Max;
            end
        end
       minss=ImL;
%       maxs=ImU;
end


function maxss=maxs(img_n)
        img_n=im2double(img_n);
        [r c]=size(img_n);
        ImL=zeros(r,c);
        ImU=zeros(r,c);
        for i=2:r-1
            for j=2:c-1
                M=img_n(i==1:i+1,j-1:j+1);
%                Min=min(M(:));
                Max=max(M(:));
                ImU(i,j)=Max;
            end
        end
       maxss=ImU;
end
function ret_img=waf(img_n)
% Mask Definition
f=1/16*[1,2,1;2,4,2;1,2,1];
% Apply filter2 function
de_noi=filter2(f,img_n);
         de_noi=uint8(de_noi)
       ret_img=de_noi;
end

function ret_img=cnn(img_n)
noisyR=img_n(:,:,1);
noisyG=img_n(:,:,2);
noisyB=img_n(:,:,3);
net=denoisingNetwork('dncnn');
denoisedR=denoiseImage(noisyR,net);
denoisedG=denoiseImage(noisyG,net);
denoisedB=denoiseImage(noisyB,net);
denoisedRGB=cat(3,denoisedR,denoisedG,denoisedB);
ret_img=denoisedRGB
end

function ret_img=median(img_n)
ret_img=medfilt2(img_n);
end
```

INPUT IMAGE

1)

2)

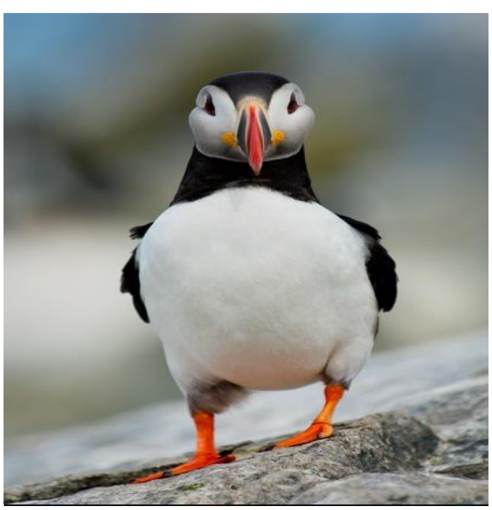
3)



4)

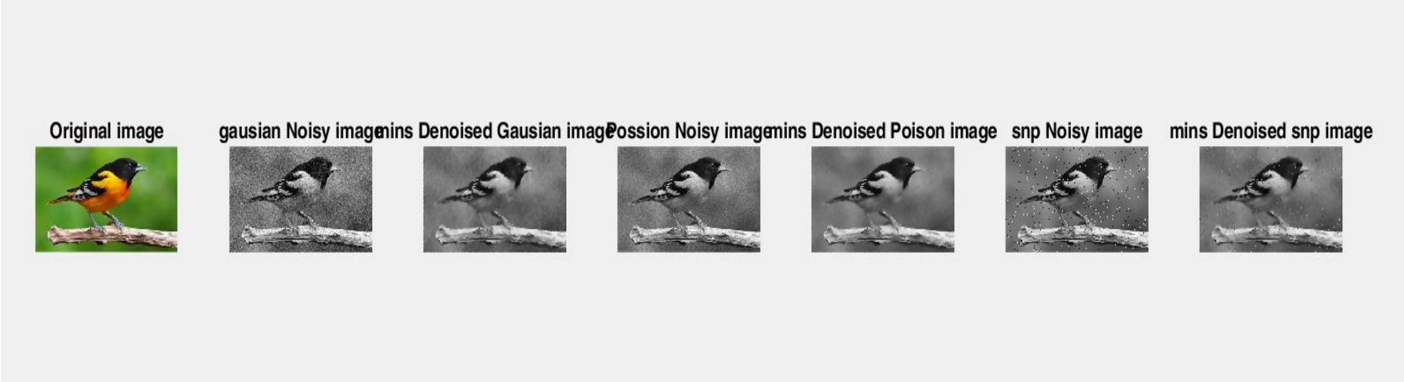
5)

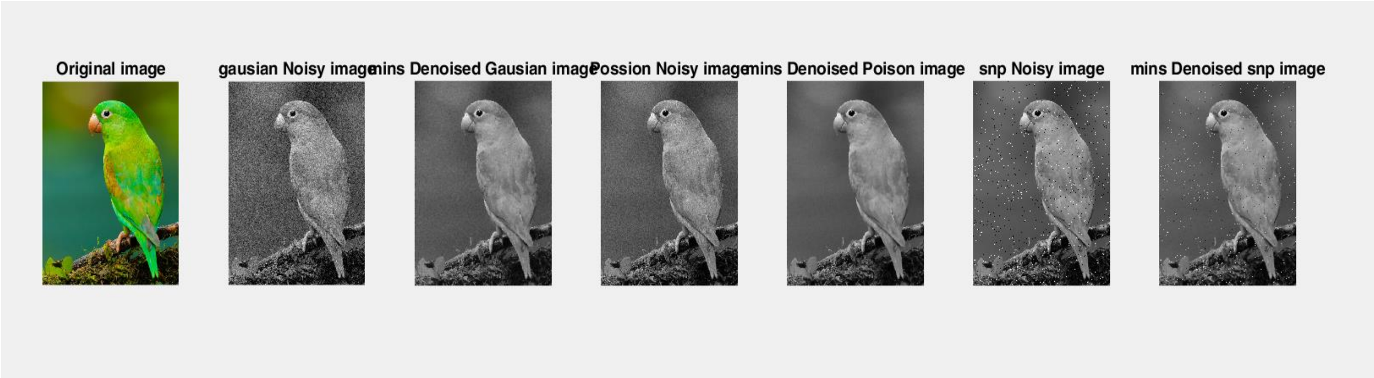
6)

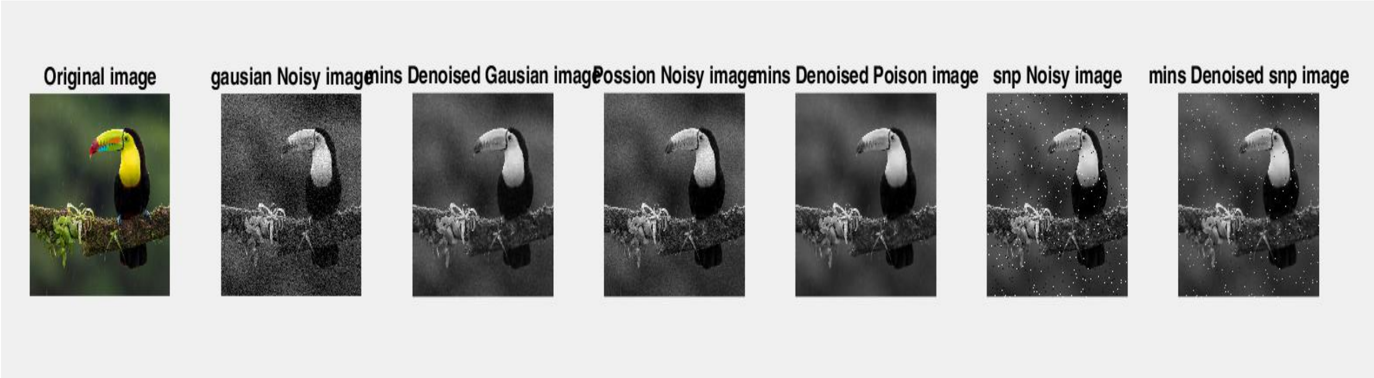


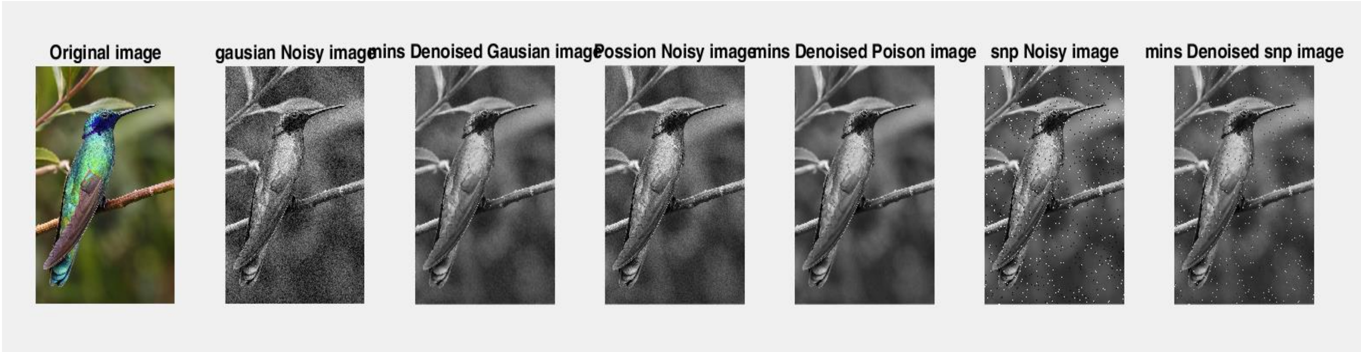
OUTPUT IMAGE

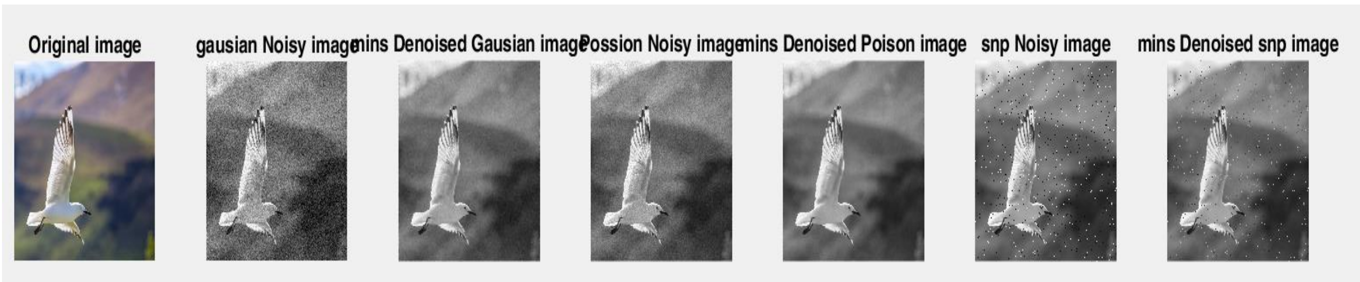
FOR wiener filter

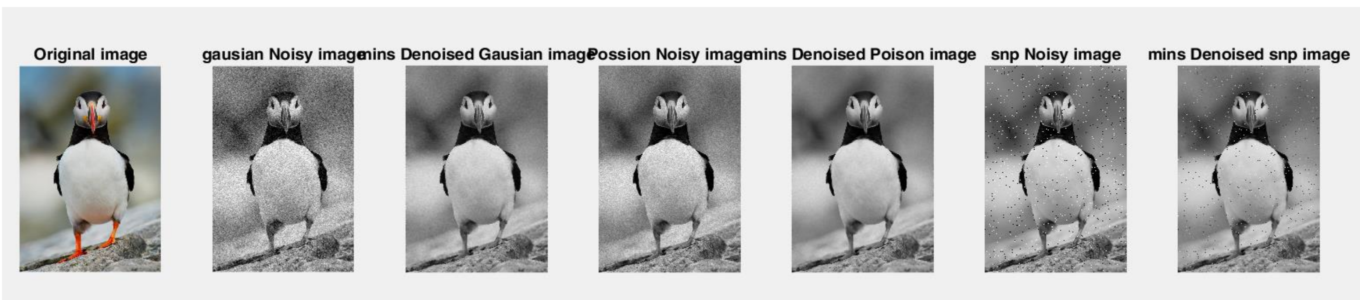




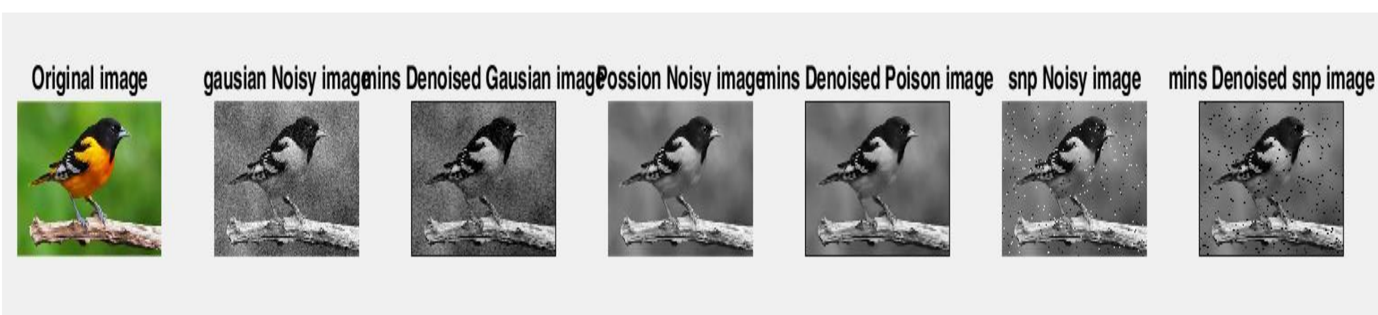


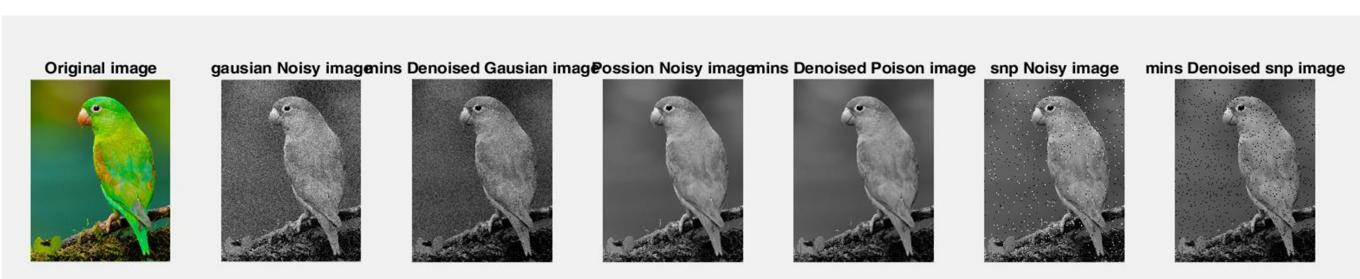


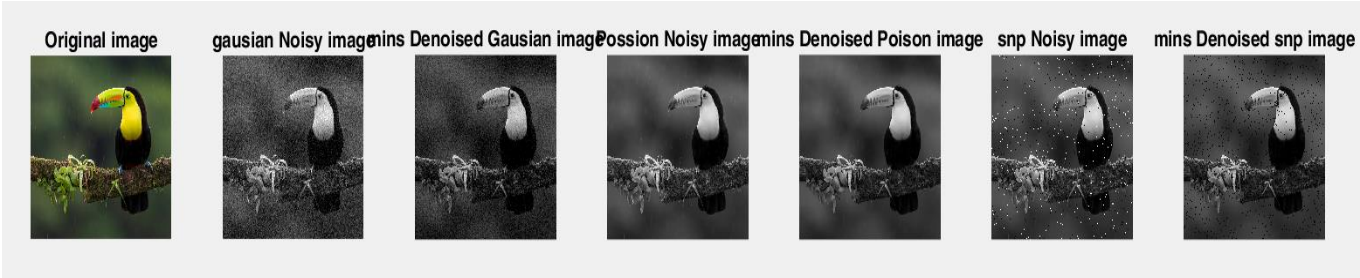




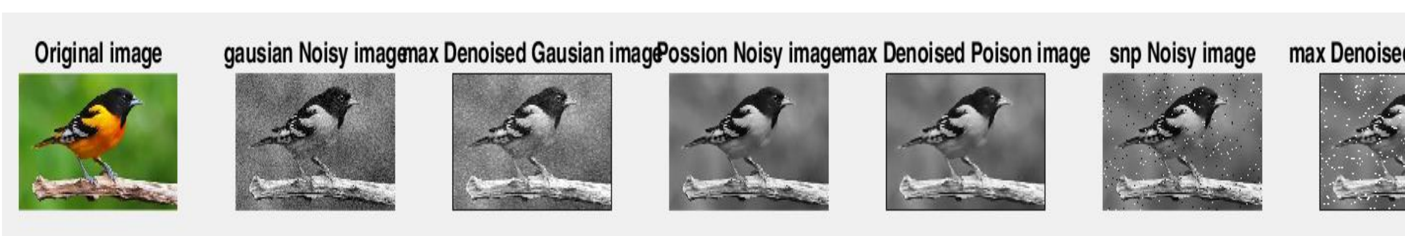
FOR MIN FILTER

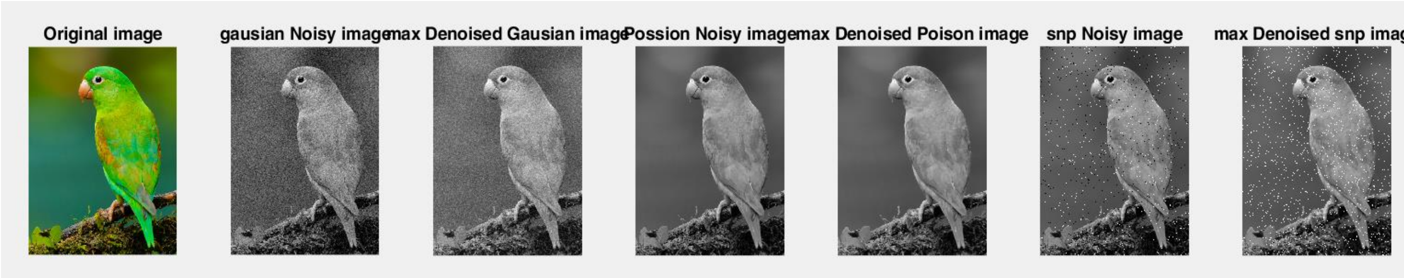


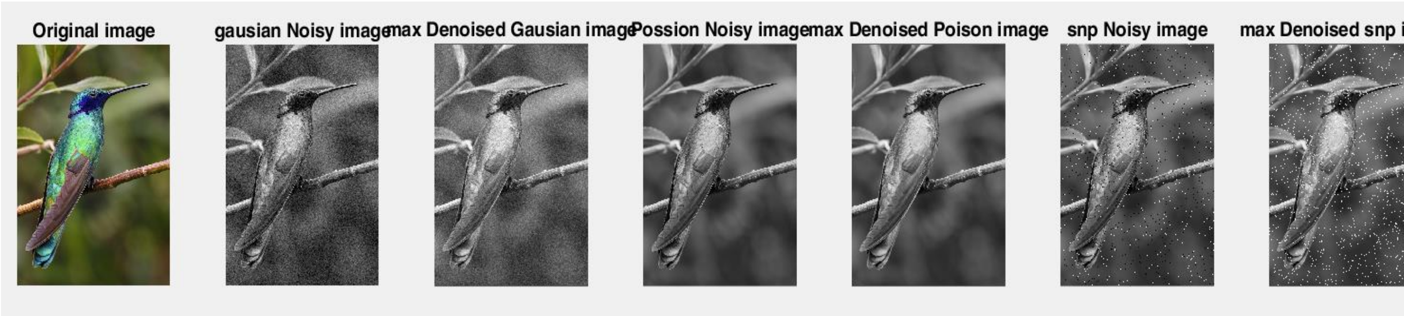


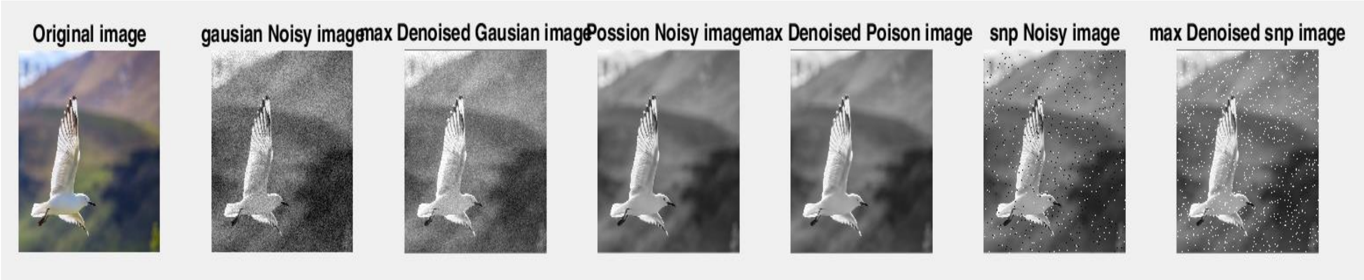


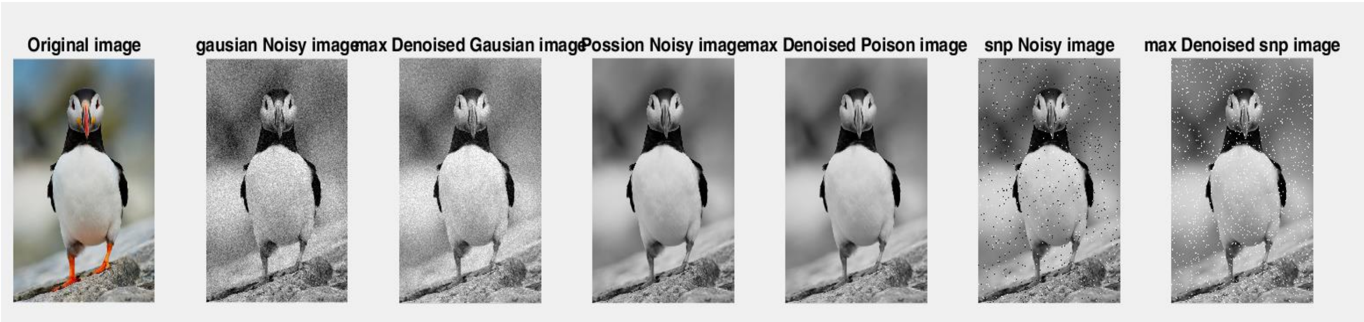
FOR MAX FILTER



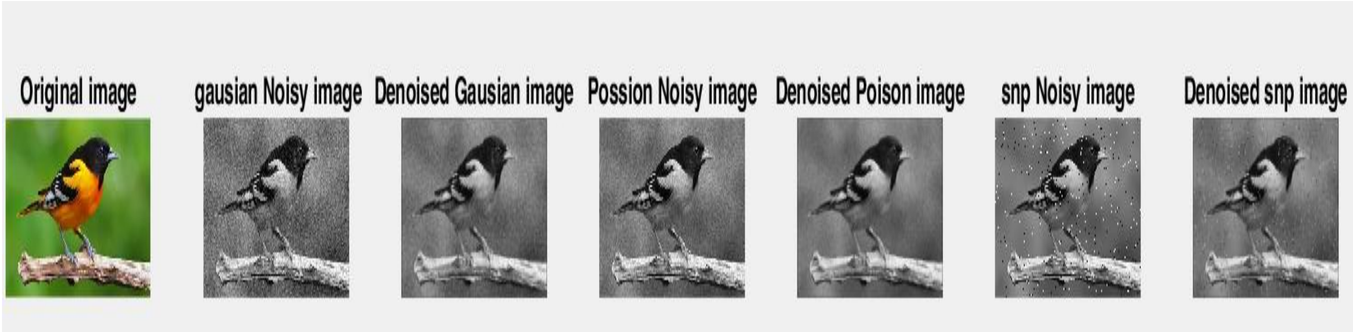


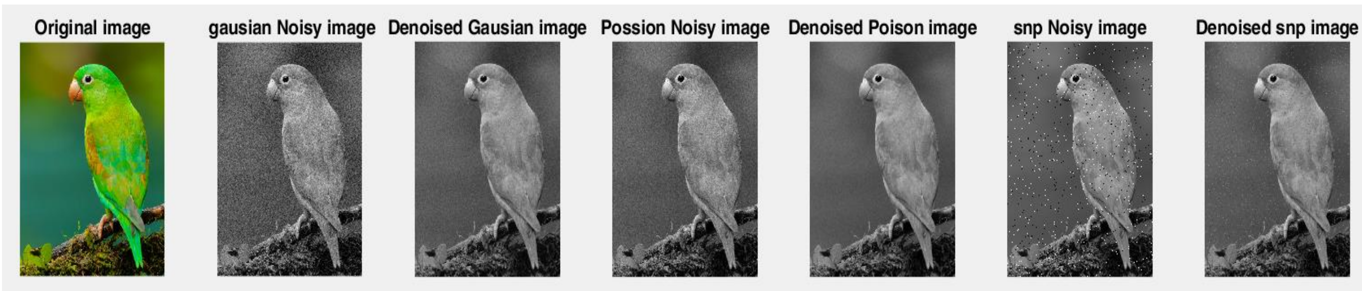


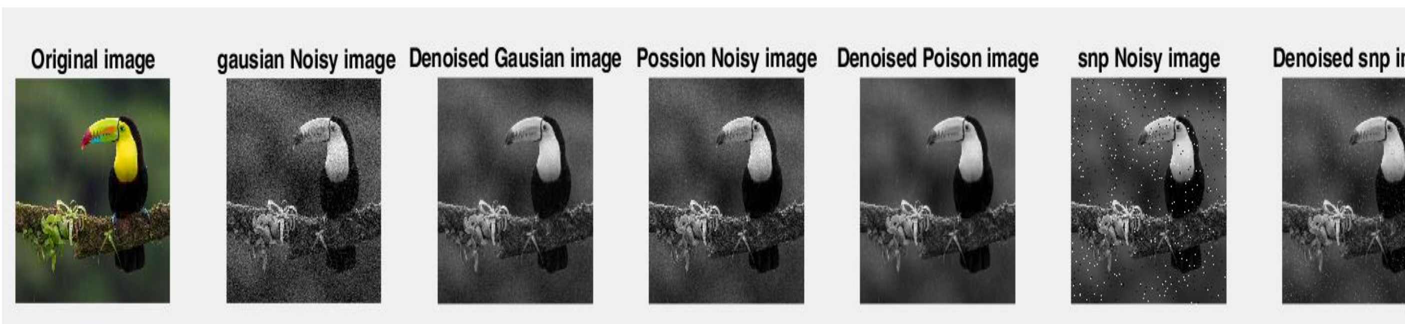


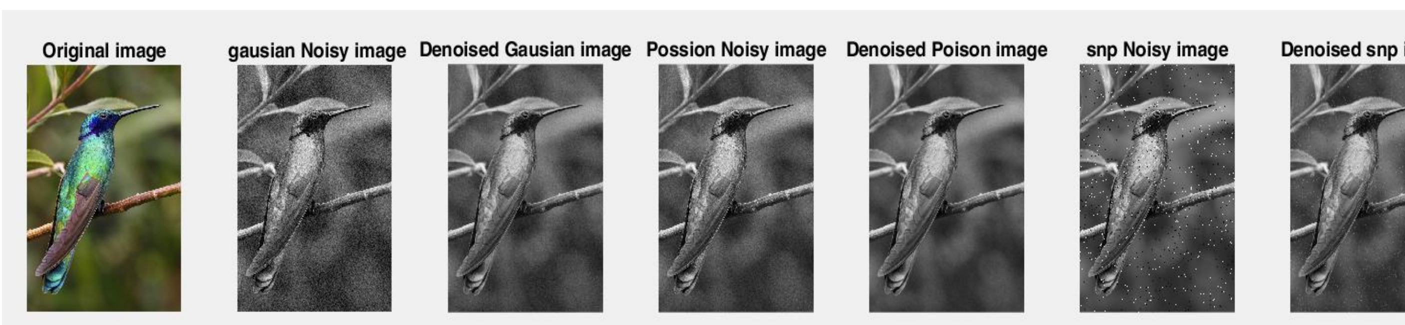


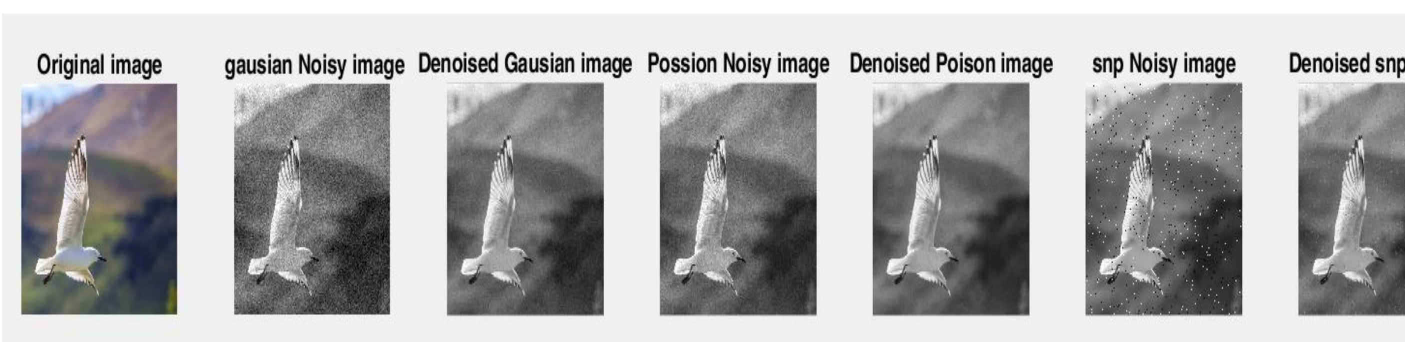
FOR WEIGHTED AVERAGE FILTER

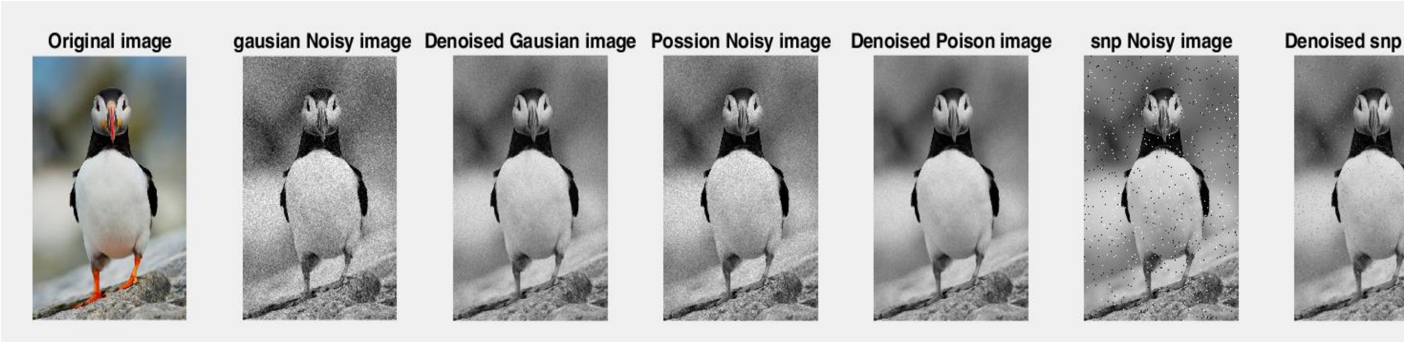

Original image
gausian Noisy image
Denoised Gausian image
Possion Noisy image
Denoised Poison image
snp Noisy image
Denoised snp image


Original image
gausian Noisy image
Denoised Gausian image
Possion Noisy image
Denoised Poison image
snp Noisy image
Denoised snp image

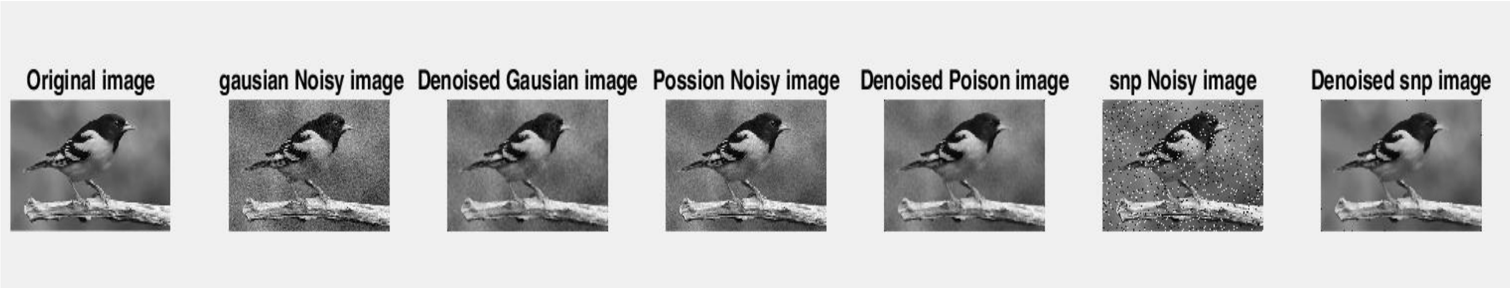

Original image
gausian Noisy image
Denoised Gausian image
Possion Noisy image
Denoised Poison image
snp Noisy image
Denoised snp

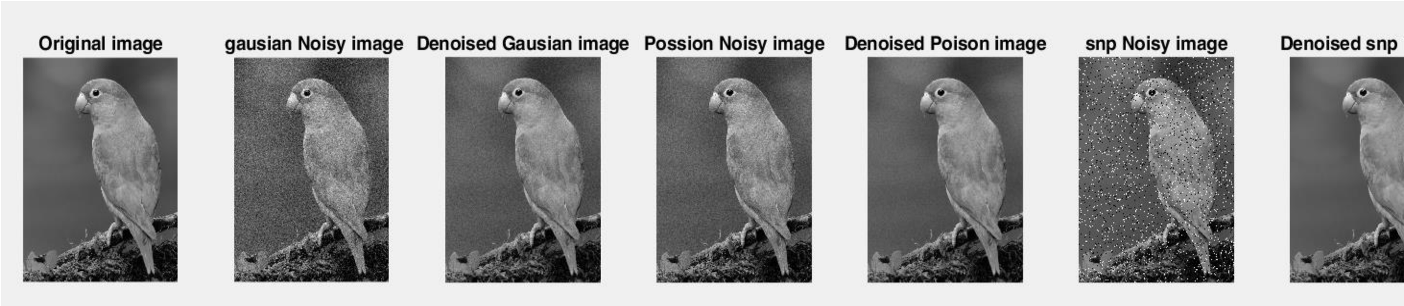

Original image
gausian Noisy image
Denoised Gausian image
Possion Noisy image
Denoised Poison image
snp Noisy image
Denoised snp

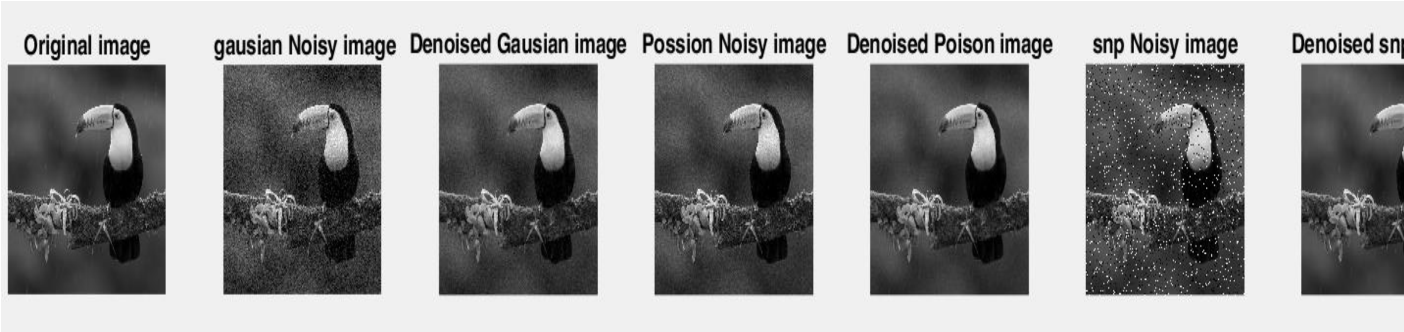

Original image
gausian Noisy image
Denoised Gausian image
Possion Noisy image
Denoised Poison image
snp Noisy image
Denoised snp

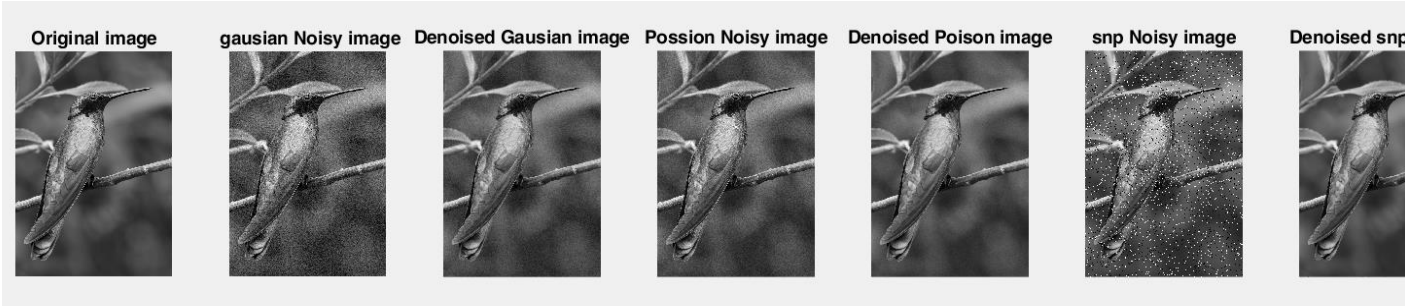


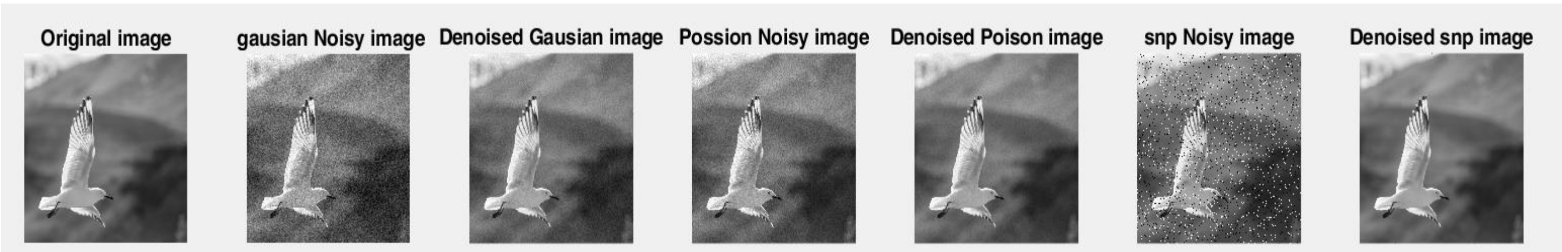
FOR MEDIAN FILTER

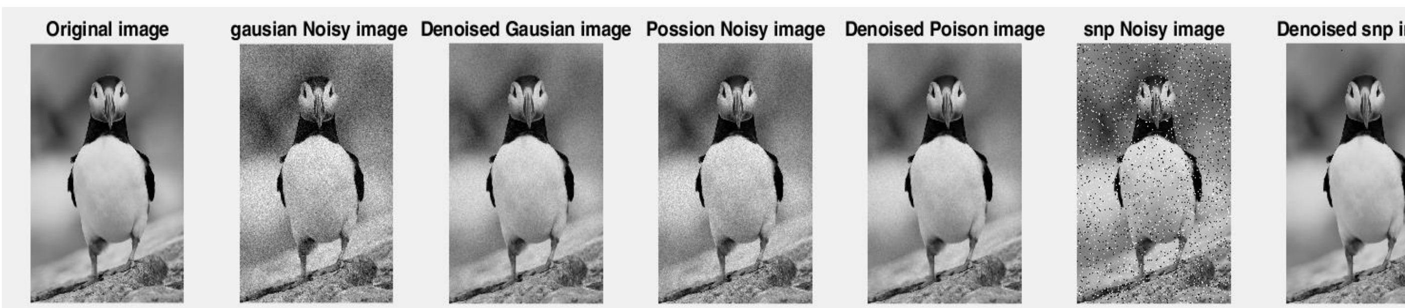




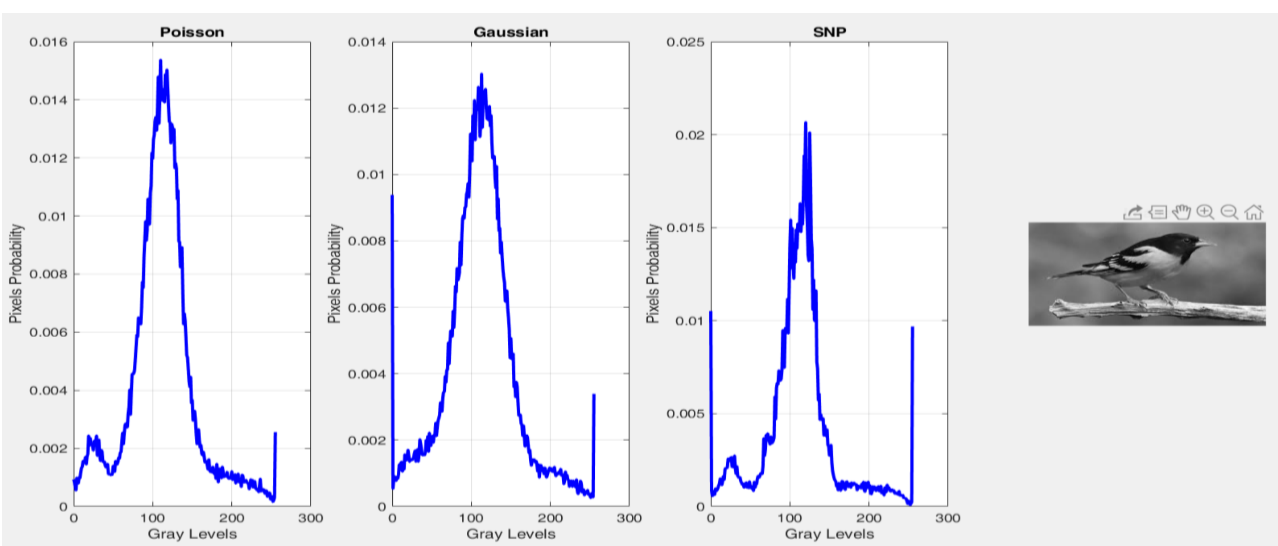


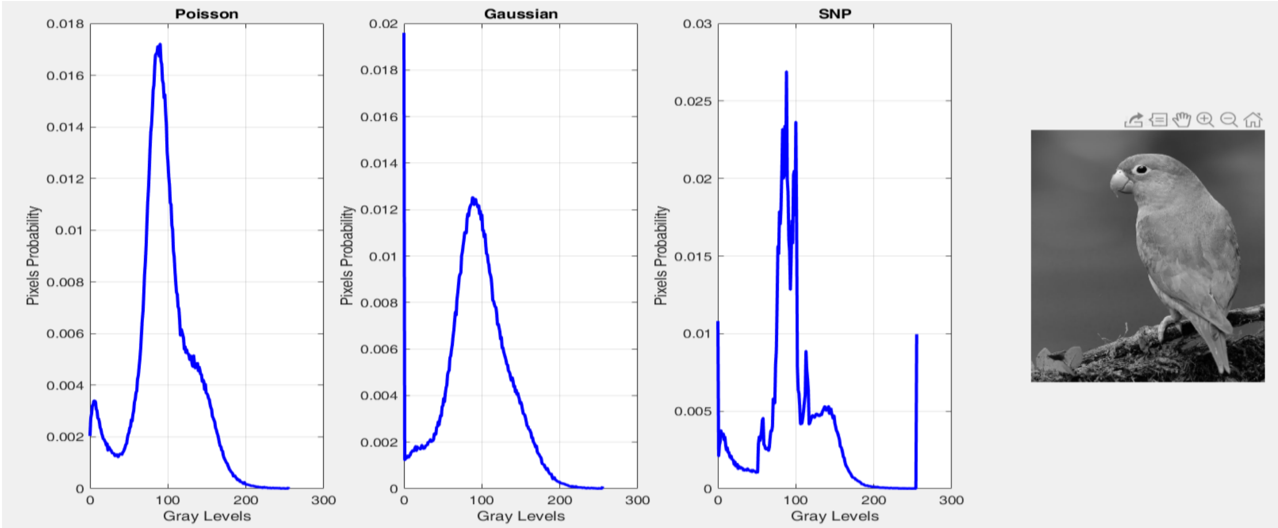

Original image
gausian Noisy image
Denoised Gausian image
Possion Noisy image
Denoised Poison image
snp Noisy image

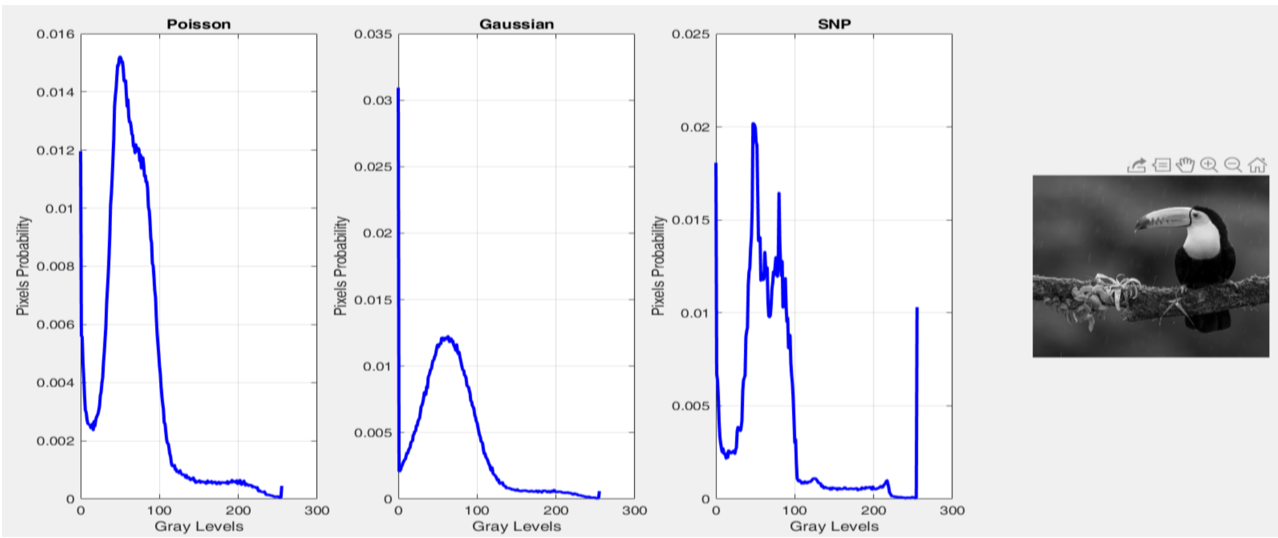

Original image
gausian Noisy image
Denoised Gausian image
Possion Noisy image
Denoised Poison image
snp Noisy image
Denoised snp image

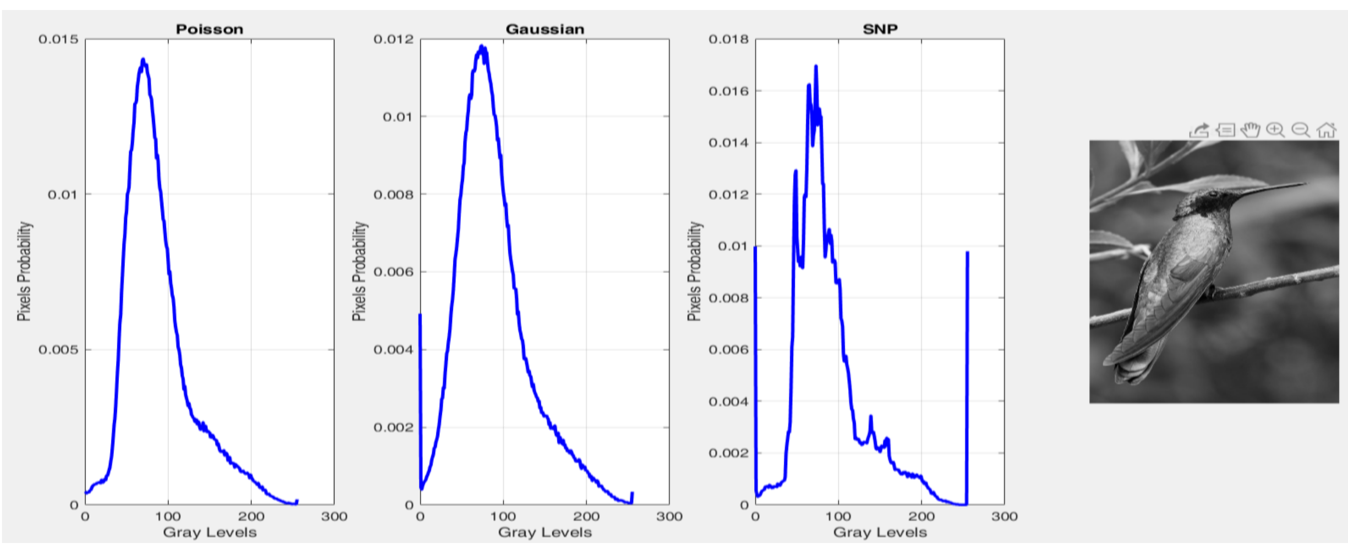

Original image
gausian Noisy image
Denoised Gausian image
Possion Noisy image
Denoised Poison image
snp Noisy image
Denoised snp

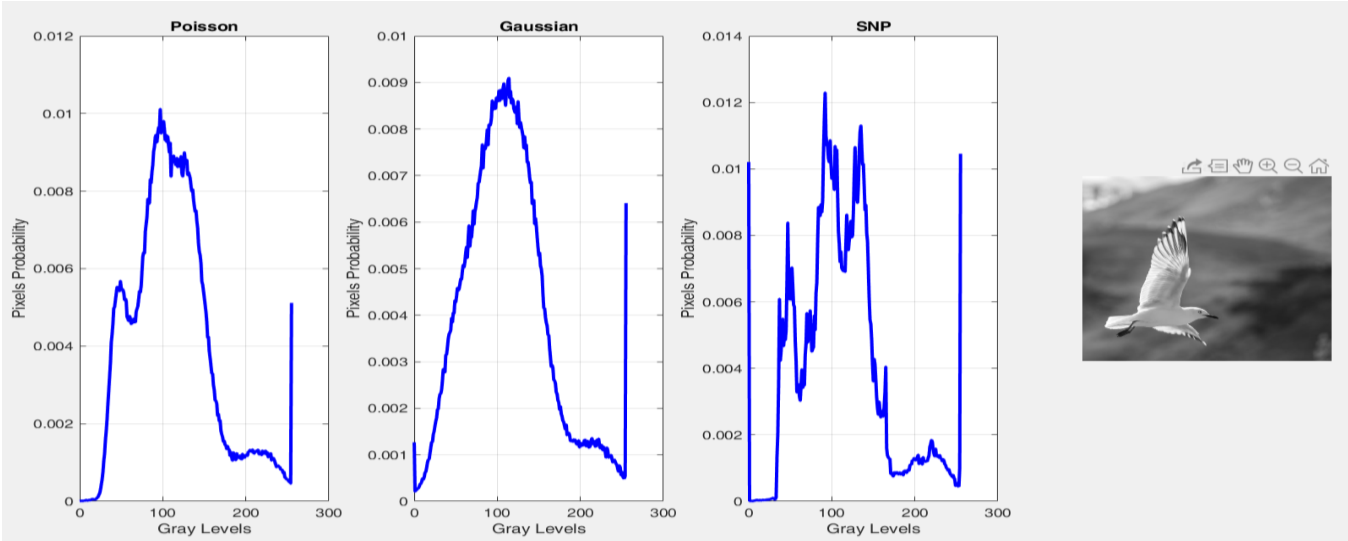
HISTOGRAM

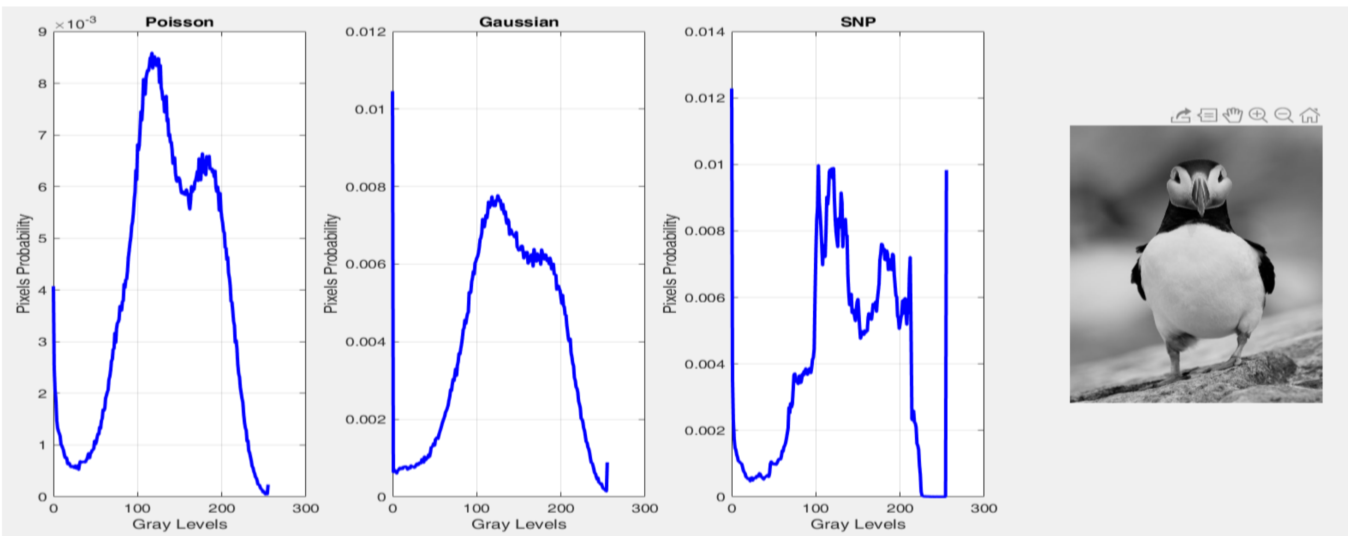

Poisson
Gaussian
SNP
Pixels Probability
Gray Levels


Poisson
Gaussian
SNP
Pixels Probability
Gray Levels


Poisson
Gaussian
SNP
Pixels Probability
Gray Levels


Poisson
Gaussian
SNP
Pixels Probability
Gray Levels





LITERATURE SURVEY

- Dr. Zhijun Pei et al [1] proposed that there are several techniques for noise removal which are well established in the field of colour image processing. The type of noise occurring in the system influences the technique of noise removal used to a great extent. In the field of image noise reduction several linear and non-linear filtering methods have been proposed. Linear filters are not able to effectively eliminate impulse noise as they have a tendency to blur the edges of an image. On the other hand nonlinear filters are suited for dealing with impulse noise. He discusses the various types of noise that can be present in an image in a detailed manner. He then goes on to discuss the noise removal techniques focussing on the Weiner filter which is a linear filter and on the Median filter which is a non-linear filter.

- Hui Ying Khaw et al [2] presented a model to effectively recognise image noise of different types and levels: impulse, Gaussian, Speckle and Poisson noise, and a mixture of multiple types of the noise. To classify image noise type, the convolutional neural network (CNN) method with backpropagation algorithm and stochastic gradient descent optimisation techniques are implemented. In order to reduce the training time and computational cost of the algorithm, the principal components analysis (PCA) filters generating strategy is deployed to obtain data adaptive filter banks. The authors validated their designed CNN with PCA for noise types recognition model with degraded images containing noise of single and combination of multiple types, with a total of 11,000 and 1650 datasets for training and testing purposes, respectively.

Lanjuan Li et al [3] proposed a wavelet domain based Convolutional Neural Network (CNN) for denoising of Medical Images. Unwanted noise is often introduced in Medical Images due to interference from the environment and equipment during acquisition, conversion, and transmission, resulting in degradation. The authors validate that their convolutional neural network structure for medical image denoising - deep neural network based on wavelet domain (deep wavelet denoising net, DWDN) exhibits high effectiveness in general medical image denoising tasks and is more excellent in the details of image.. The deep wavelet denoising net (DWDN) a CNN based on wavelet transform takes a noise image(NI) as its input using 2D Haar Transform and the 2D Haar Transform of original Image (OI) as label to train the neural network.

PROPOSED METHODOLOGY

Reference	***Methods Used***	***Evaluation***	***Merits and Demerits***
Image Noise Reduction and Filtering Techniques Author- Dr. Zhijun Pei et al	• Linear Filter(weiner filter) • Non-linear filter(median filter)	Discusses different types of noises present in image, discusses how linear filter(stress on weiner filter) and no- linear filter(stress on median filter) used for the removal of noise	Merits: • Discusses how the results obtained using median filter technique ensures noise free and quality of the image as well. • Weiner filter Preserves edges and other high frequency parts of an image Demerits: • Focuses only on the median and adaptive(weiner) filters of

			noise removal ● No common basis of comparison to prove which filter is better suitable for which noise
Image noise types recognition using convolutional neural network with principal components analysis Author - Hui Ying Khaw et al	● the Convolutional Neural Network (CNN) method with backpropagation algorithm and stochastic gradient descent optimisation technique ● Principal Components Analysis (PCA) filters	Effectively studies and proposes a model to recognise different type of noise present in images like Gaussian noise, Speckle noise and Poisson noise, etc. Classifies the image noise type using the convolutional neural network (CNN) method	Merits: ● The variety and complexity used in this paper is unprecedented as it hasn't been seen in any previous research work ● The capability of the intelligent system model that the authors developed in handling images that

			have been degraded has surpassed human-eye performance in the recognition of noise types • The proposed model has an accuracy of 99.3% in noise recognition and can recognise 8 different types of noise • Demerits: • Just identifies and classifies different types of noise without proposing the ways of removing the said noises.

CNN denoising for medical image based on wavelet domain Authors - Lanjuan Li, Jingyang Wu, Xinyu Jin et al	• A new Convolutional Neural Network Structure based on wavelet domain specially for medical images called deep wavelet denoising net (DWDN). • Uses 2D Haar Transform for taking images as input	Shows an effective method for denoising medical images using deep wavelet denoising net. The proposed method gives better results than Deep CNN (DnCNN) and Collaborative filtering for denoising images.	Merits: • Overcomes the need of manually choosing parameters like in case of other CNNs. • It simplifies difficulty for CNN to learn object, and improves performance. • In terms of PSNR performs better than DnCNN. Demerits: • The model gives better performance for details of image. Gives similar performance for general image. • Focuses mainly on

			the CNN filtering technique of noise removal. Other filters are not discussed

RESULT

→ **PSNR** → **SSIM**

IMAGE 1

Noises	SnP	Gaussian	Poisson	SnP	Gaussian	Poisson
Weiner	25.88328	27.41639	30.10023	0.761984	0.793249	0.883066
Min Filter	19.14048	19.35038	22.50224	0.592321	0.438819	0.86916
Max Filter	17.90058	19.28274	22.5066	0.545106	0.439655	0.871541
Weighted avg Filter	26.02288	26.41762	27.38771	0.71717	0.68508	0.802855
CNN	22.79673	31.0871	44.69257	0.79811	0.976045	0.998804
Median	27.09646	25.61644	26.80176	0.869152	0.655525	0.79811

IMAGE 2

Noises	SnP	Gaussian	Poisson	SnP	Gaussian	Poisson
Weiner	25.57942	30.09363	34.06648	0.650943	0.77059	0.895591
Min Filter	22.08744	21.22702	27.88003	0.608866	0.373949	0.907713
Max Filter	18.5101	21.00222	27.84002	0.515541	0.374288	0.911993
Weighted avg Filter	29.4671	30.06121	33.10662	0.739875	0.690132	0.852653
CNN	22.23121	32.32053	50.48492	0.824531	0.970709	0.999237
Median	33.90317	29.06111	32.09896	0.834158	0.645297	0.824531

IMAGE 3

Noises	SnP	Gaussian	Poisson	SnP	Gaussian	Poisson
Weiner	24.39589	30.17217	34.56331	0.632873	0.7811	0.922885
Min Filter	23.46627	21.21626	27.36497	0.667475	0.362532	0.909921
Max Filter	17.32587	20.88522	27.29885	0.487779	0.348022	0.915914
Weighted avg Filter	28.65478	29.66449	32.86141	0.729531	0.670422	0.880941
CNN	21.85701	32.92565	52.47678	0.49446	0.922582	0.998923
Median	32.7779	28.6652	31.86751	0.858122	0.631641	0.858122

IMAGE 4

Noises	SnP	Gaussian	Poisson	SnP	Gaussian	Poisson
Weiner	25.26747	30.42975	34.46902	0.662952	0.809868	0.929009
Min Filter	21.95574	20.75083	26.64893	0.616532	0.358527	0.912063
Max Filter	18.09085	20.75114	26.61644	0.508077	0.365691	0.914239
Weighted avg Filter	28.88839	29.44921	32.04125	0.73739	0.680531	0.860136
CNN	22.56605	32.70404	44.97473	0.745212	0.976661	0.999328
Median	32.26738	28.45677	31.08863	0.836075	0.638019	0.836075

IMAGE 5

Noises	SnP	Gaussian	Poisson	SnP	Gaussian	Poisson
Weiner	26.02792	32.28156	36.96062	0.669683	0.815305	0.940535
Min Filter	20.51713	20.68159	26.4514	0.58262	0.319941	0.952207
Max Filter	19.01455	20.74618	26.47264	0.534668	0.322275	0.95176
Weighted avg Filter	29.84959	30.44274	33.57842	0.723776	0.657689	0.838697
CNN	25.40679	36.59183	60.23176	0.640736	0.951232	0.999846
Median	37.81126	30.1351	33.99844	0.825605	0.628321	0.825605

IMAGE 6

Noises	SnP	Gaussian	Poisson	SnP	Gaussian	Poisson
Weiner	25.86642	31.25259	35.04309	0.67591	0.812043	0.917689
Min Filter	19.11107	20.60895	25.92933	0.549377	0.357263	0.939282
Max Filter	20.03783	20.53742	25.92705	0.586511	0.352117	0.939773
Weighted avg Filter	29.66439	30.17705	32.59651	0.738258	0.675175	0.812709
CNN	25.39807	34.57273	56.94531	0.609129	0.921529	0.999566
Median	35.91907	29.67822	32.48425	0.791702	0.644256	0.791702

Conclusion

Convolutional Neural Network is the best technique that can be used to remove noises present in an image. A further research in Neural Networks can yield better results in image noise removal techniques. Moreover, Machine Learning and Artificial Intelligence can be used with training sets to make a model which can directly identify the type of noise present in an image. This model can then by employed to find out the best Noise removal Techniques based upon the PSNR, SSIM values of the denoised images corresponding to the filters used in the training data set.

Over all Discussion

The PSNR & SSIM Comparison Table shows the following:

1. For salt and pepper noise median filter is the best technique.

2. For Gaussian noise CNN is the best technique.

3. For Poisson noise CNN is the best technique.

Further it can be seen that Convolutional Neural Network gives comparatively high PSNR & SSIM Values. This means that this technique is useful for removing all kinds of noise present in the image.

FUTURE WORK

Tabular Comparison With existing Work

Previous Works	Proposed Work
• Dr. Zhijun Pei et al [1] discusses only median and wiener filter for noise removal techniques. • No common basis of comparison used to find out which filter is better	• discusses Min/ Max Filter, Median Filter, Convolutional Neural Network (CNN) Filter, Weighted Average Filter, Weiner Filter for noise removal. • PSNR is used for comparing various noise removal techniques. • SSIM is used for measuring the similarity between two images
• Hui Ying Khaw et al [2] identifies and classifies different types of noise	• Classify noise using histogram and apply suitable filters on the image.
• Lanjuan Li et al [3] focuses on Dn CNN for noise removal. • Gives better result for details in images	• Gives good result for denoising general images

REFERENCE

[1] Abdalla Mohamed Hambal, Dr. Zhijun Pei and Faustini Libent Ishabailu (2017). Image Noise Reduction and Filtering Techniques. International Journal of Science and Research (IJSR), Volume 6 Issue 3.

[2] Hui Ying Khaw , Foo Chong Soon, Joon Huang Chuah and Chee-Onn Chow (2017). Image noise types recognition using convolutional neural network with principal components analysis. IET Image Processing Journal Volume 11 Issue 12, pages: 1238 – 1245

[3] Lanjuan Li, Jingyang Wu and Xinyu Jin (2018). CNN Denoising for Medical Image Based on Wavelet Domain. 9th International Conference on Information Technology in Medicine and Education (ITME). IEEE.

Appendix for Individual Contribution Details in the Group

S. No. Digital	Reg No	Role and Responsibility	Signature
1	19BCE2002	Synopsis, graphical abstract, introduction, background of project, Code to plot PDF of Noise Images, Code for Noise Removal Filters, code to plot the histograms of the noises, Evaluation and Result Analysis, PSNR Comparison Table, Appendix for Acronyms, Overall Discussion, Conclusion, Power Point Presentation for the Demonstrations	KARTIK GOEL
2	19BCE2067	Introduction, Literature survey, background of project, Code for Noise Removal Filters, Evaluation and Result Analysis, SSIM Comparison Table, Tabular Comparison With existing Work, Overall Discussion, Conclusion, References, Appendix for Acronyms	TANISHQ SHAH
3	19BCE0987	Introduction, Literature survey, background of project, Code for Noise Removal Filters, Evaluation and Result	VANSH AGGARWAL

		Analysis, SSIM Comparison Table, Tabular Comparison With existing Work, Overall Discussion, Conclusion, References, Appendix for Acronyms	