```
%%                           Bilateral Filtering
%   V Khagesh Kumar
%   170102070
%   ECE

clear;

%% Bilateral Filter with 3 examples
% Load Image for spnoisy
mask = 5;
file_name = "spnoisy.jpg";
original_image = rgb2gray(imread(file_name));
image = double(original_image);
sigma_r = 80;
sigma_d = 10;
bilateral_filter(mask, original_image, image,sigma_r,sigma_d,file_name);

% Load Image for spunifnoisy
mask = 5;
file_name = "spunifnoisy.jpg";
original_image = rgb2gray(imread(file_name));
image = double(original_image);
sigma_r = 70;
sigma_d = 10;
bilateral_filter(mask, original_image, image,sigma_r,sigma_d,file_name);


% Load Image for unifnoisy
mask = 5;
file_name = "unifnoisy.jpg";
original_image = rgb2gray(imread(file_name));
image = double(original_image);
sigma_r = 20;
sigma_d = 10;
bilateral_filter(mask, original_image, image,sigma_r,sigma_d,file_name);



%% Bilateral Filtering Function
function bilateral_filter(mask, original_image, image,sigma_r,sigma_d, file_name)

    [m,n] = size(image);
    image = padarray(image,[floor(mask/2),floor(mask/2)],'replicate');

    %Domain Filter
    W_d = zeros(mask,mask);
    for i=1:mask
        for j=1:mask
            W_d(i,j) = exp( -(sum(abs([i,j]-[ceil(mask/2),ceil(mask/2)]))^2)/(2*sigma_d*
        end
    end

    % Range filter
    final_image = original_image;
    for i=1:m
        for j=1:n
            image_section = image(i:i+mask-1,j:j+mask-1);
            %image_section = image(i:i+4,j:j+4);
            W_r = range_filter(image_section,sigma_r,mask);
            final_image(i,j) = round(sum(sum(W_d.*W_r.*image_section))/sum(sum(W_d.*W_r)
        end
    end

    final_image = uint8(final_image);

    % Plot
    figure();
    imshow(original_image);
    title(strcat('Original Image : ',file_name));
    figure();
    imshow(final_image);
    title(strcat('Reduced Noise Image : ',file_name));
    imwrite(original_image,strcat('gray_',file_name),'JPG');
    imwrite(final_image,strcat('output_',file_name),'JPG');
end


%% Function
function result = range_filter(image,sigma_r,mask)
    result = exp( -((image-image(ceil(mask/2),ceil(mask/2))).^2)/(2*sigma_r*sigma_r) );
    %result = exp( -((image-image(3,3)).^2)/(2*sigma_r*sigma_r) );
end
```