Digital Image Processing
Project 3
Implementation of a Fuzzy Logic Image Processing System

Omar Rawashdeh
Harshal Raut
Utsav Shah

October 18, 2016

List of Figures

Introduction

Concept

Fuzzy logic is an approach to computing based on "degrees of truth" rather than the usual "true or false" (1 or 0) Boolean logic on which the modern computer is based.

History

The idea of fuzzy logic was first advanced by Dr. Lotfi Zadeh of the University of California at Berkeley in the 1960s. Dr. Zadeh was working on the problem of computer understanding of natural language. Natural language (like most other activities in life and indeed the universe) is not easily translated into the absolute terms of 0 and 1. (Whether everything is ultimately describable in binary terms is a philosophical question worth pursuing, but in practice much data we might want to feed a computer is in some state in between and so, frequently, are the results of computing.) It may help to see fuzzy logic as the way reasoning really works and binary or Boolean logic is simply a special case of it[1].

Interpretations

Fuzzy logic has two different meanings. In a narrow sense, fuzzy logic is a logical system, which is an extension of multivalued logic. However, in a wider sense fuzzy logic (FL) is almost synonymous with the theory of fuzzy sets, a theory which relates to classes of objects with unsharp boundaries in which membership is a matter of degree. In this perspective, fuzzy logic in its narrow sense is a branch of FL. Even in its more narrow definition, fuzzy logic differs both in concept and substance from traditional multivalued logical systems.

Fuzzy logic is all about the relative importance of precision: How important is it to be exactly right when a rough answer will do![2].

Problem A: Intensity Transformation

MATLAB fuzzy tool to implement and repeat experiment

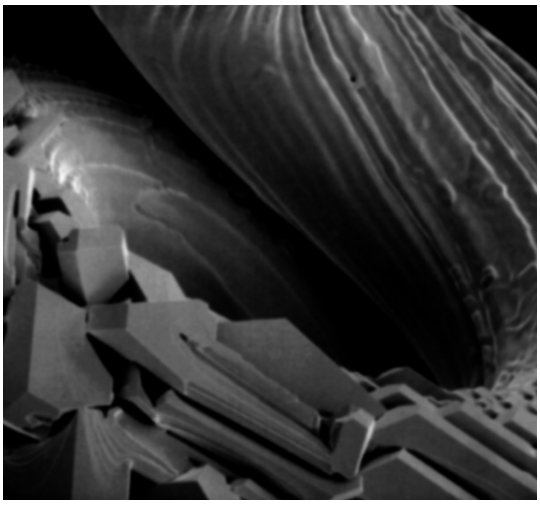

Figure 1: Original Image

Figure 2: Intensity Transformed Image, Edges=70

Comparison of result with report in text

The experiment is totally different than the histogram equalized images as the fuzzy inference system developed makes the dark pixels darker, bright pixels brighter and gray pixels in the same levels.

The Membership Function used in the experiment is the following:

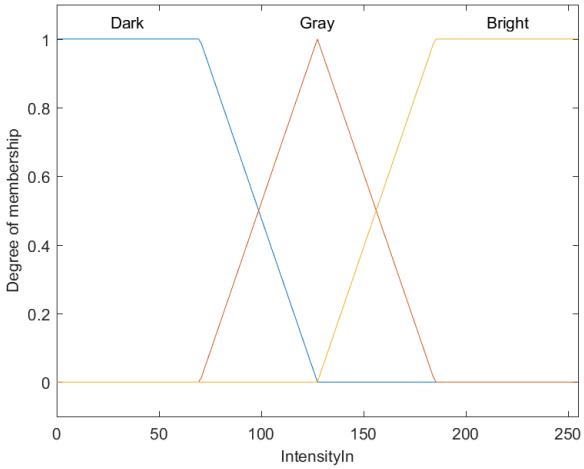

Figure 3: Membership Function

Modify the membership functions and test on more images

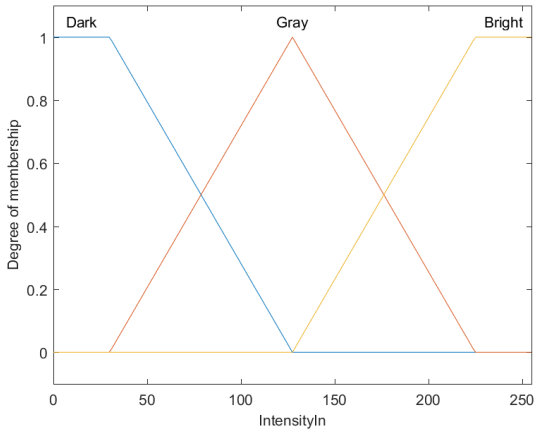

Figure 4: Membership Function, Edges=30

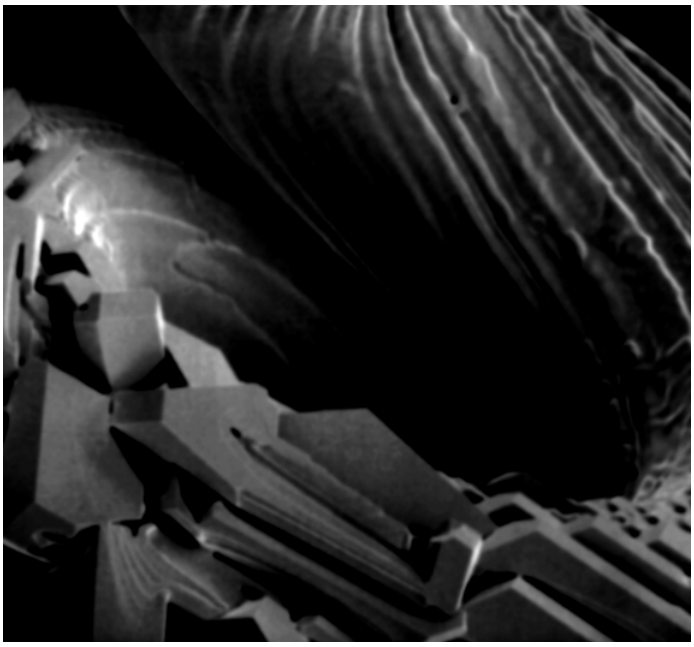

Figure 5: Intensity Transformed Image, Edges=30

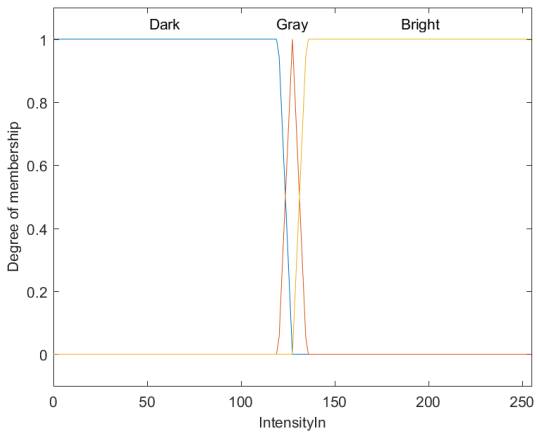

Figure 6: Membership Function, Edges=120

Figure 7: Intensity Transformed Image, Edges=120

Problem B: Spatial Filtering

MATLAB fuzzy tool to implement and repeat experiment

Figure 8: Used Image

Figure 9: Processed Image With Scaling

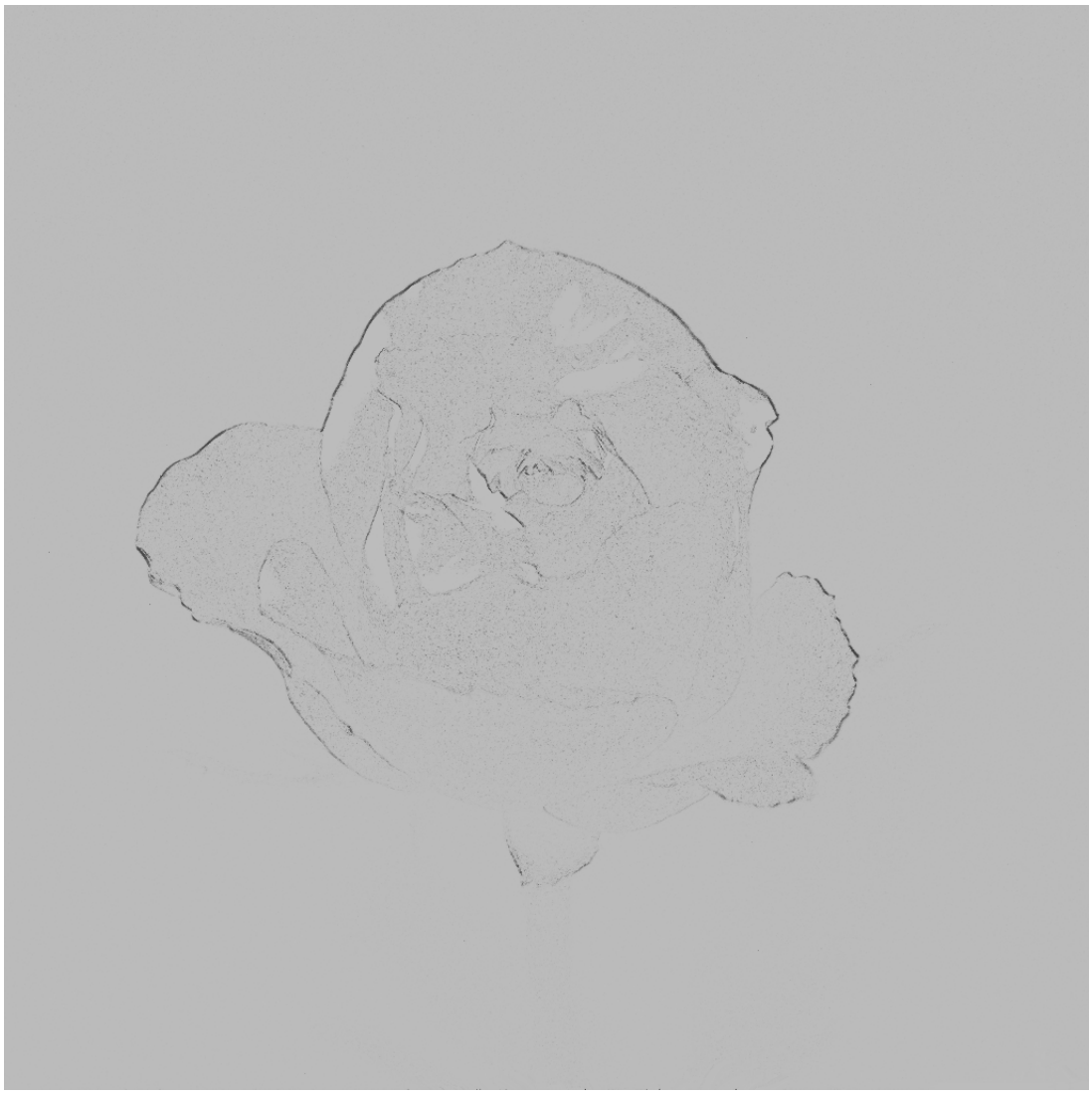

Figure 10: Processed Image Without Scaling

Comparison of result with report in text

The experiment is totally different than the histogram equalized images as the fuzzy inference system developed makes the dark pixels darker, bright pixels brighter and gray pixels in the same levels.

The Membership Functions used in the experiment is the following:

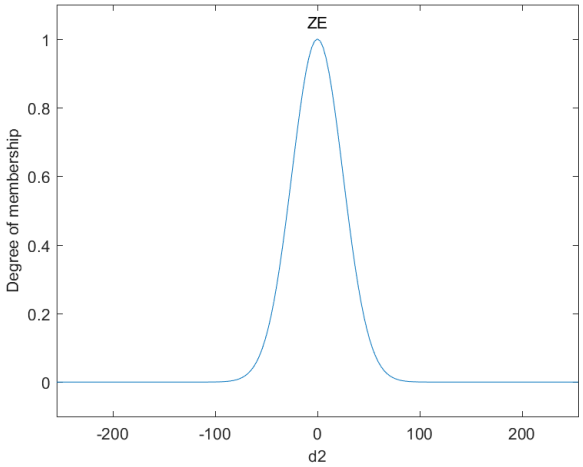

Figure 11: Input 1 Membership Function, Spread=25, d2

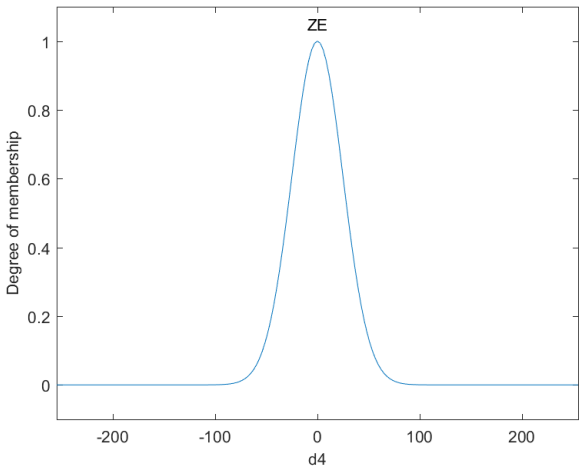

Figure 12: Input 2 Membership Function, Spread=25, d4

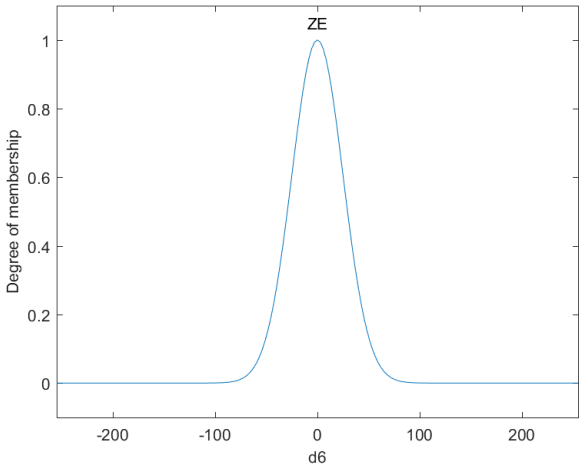

Figure 13: Input 3 Membership Function, Spread=25, d6

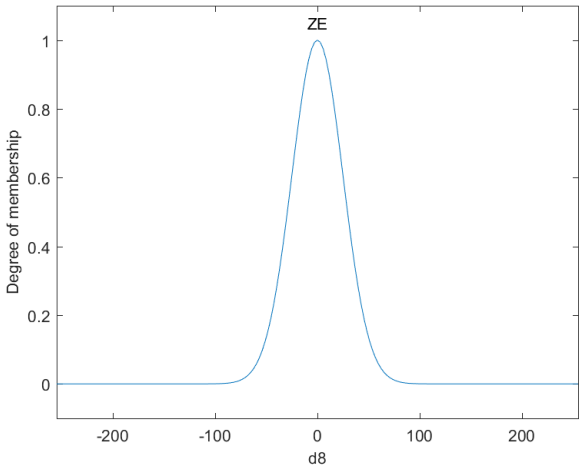

Figure 14: Input 4 Membership Function, Spread=25, d8

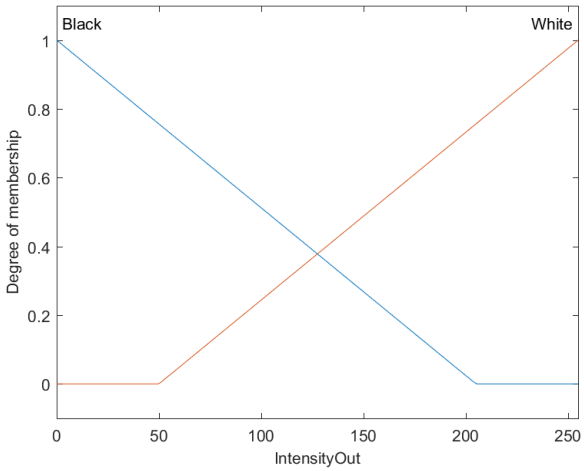

Figure 15: Output Membership Function, Spread=25

Modify the membership functions and test on more images

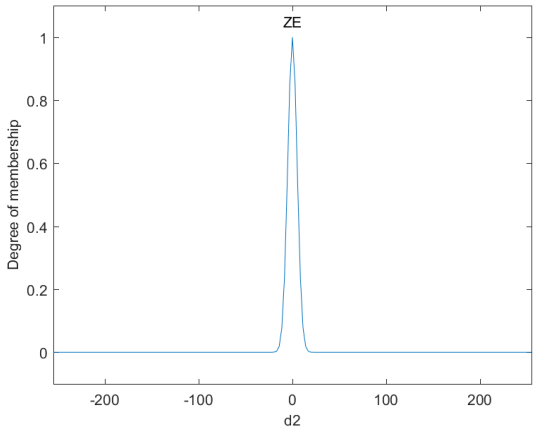

Figure 16: Input 1 Membership Function, Spread=5, d2

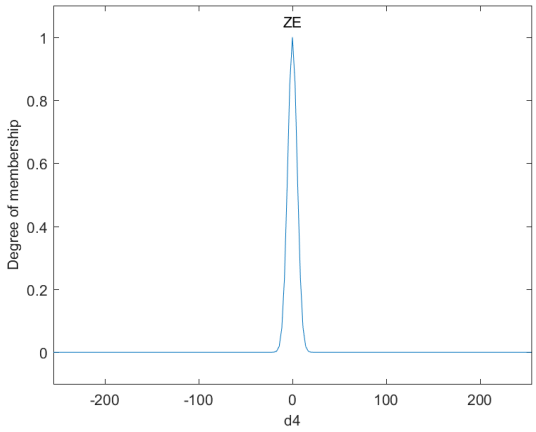

Figure 17: Input 2 Membership Function, Spread=5, d4

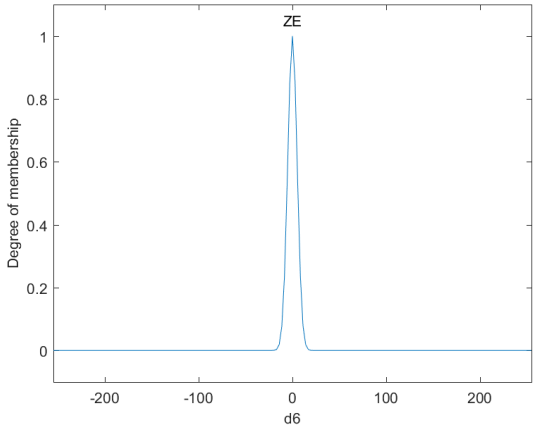

Figure 18: Input 3 Membership Function, Spread=5, d6

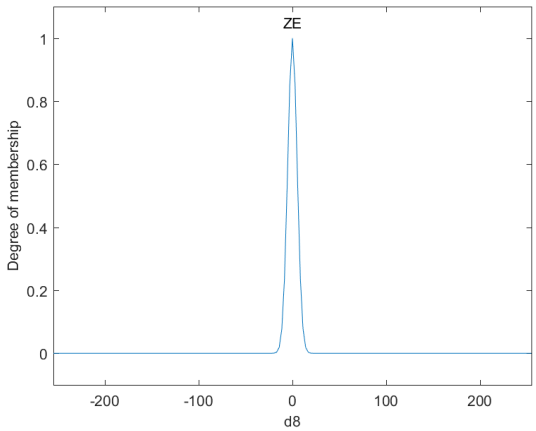

Figure 19: Input 4 Membership Function, Spread=5, d8

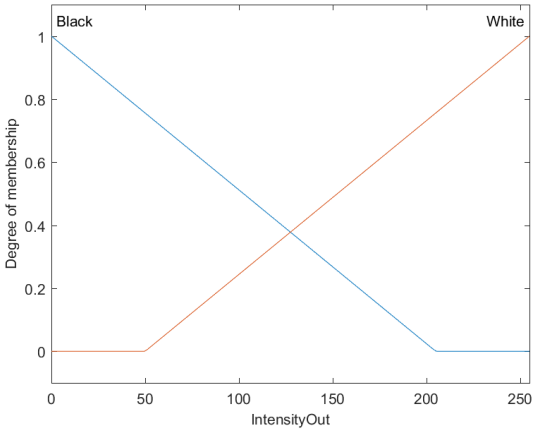

Figure 20: Output Membership Function, Spread=5

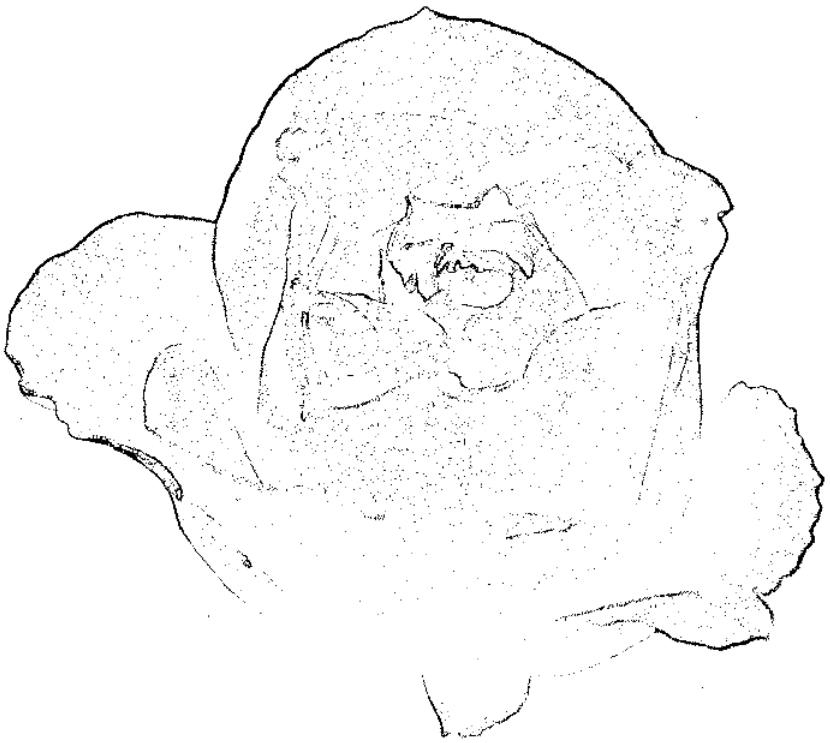

Figure 21: Processed Image With Scaling, Spread=5

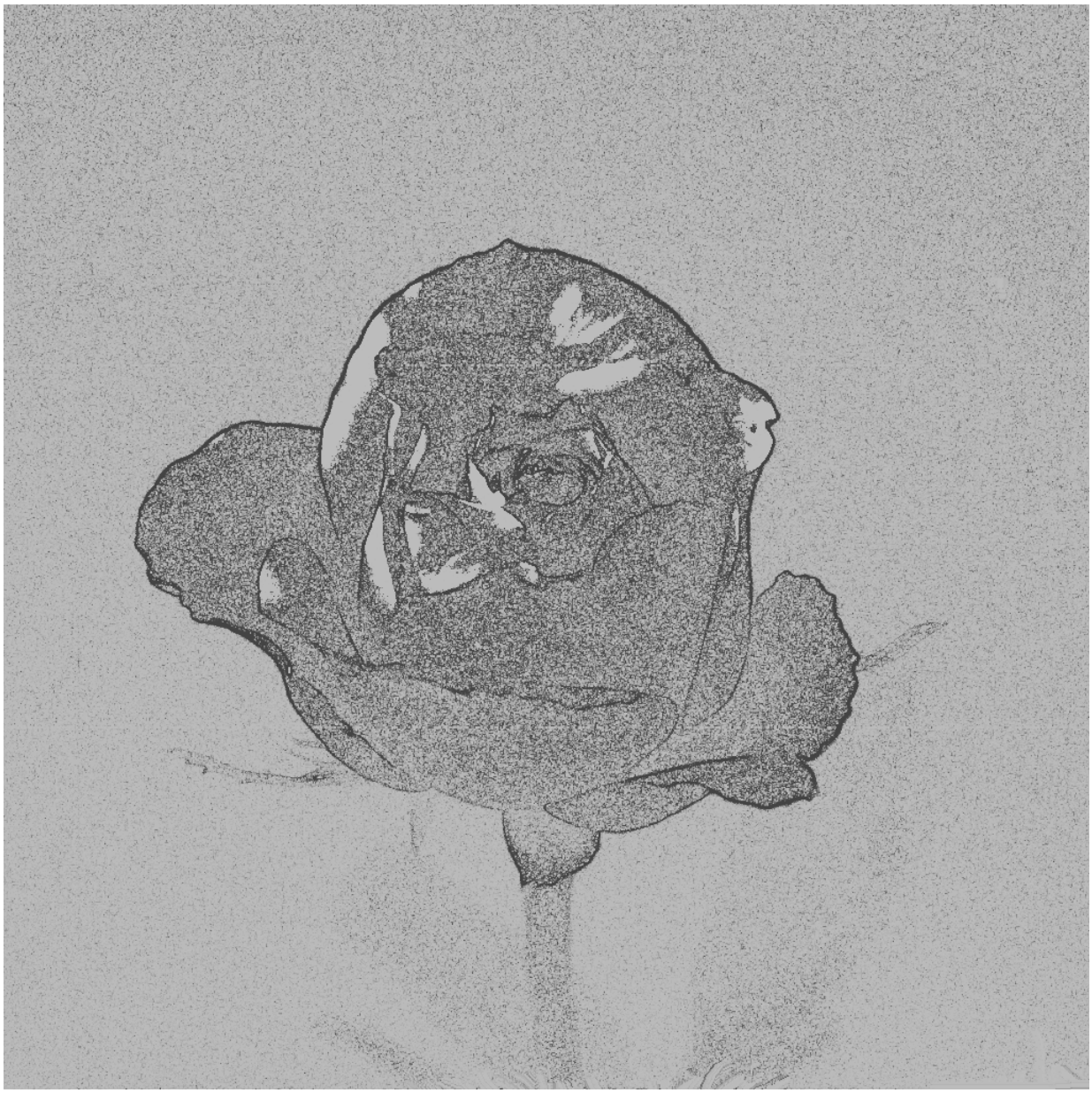

Figure 22: Processed Image Without Scaling, Spread=5

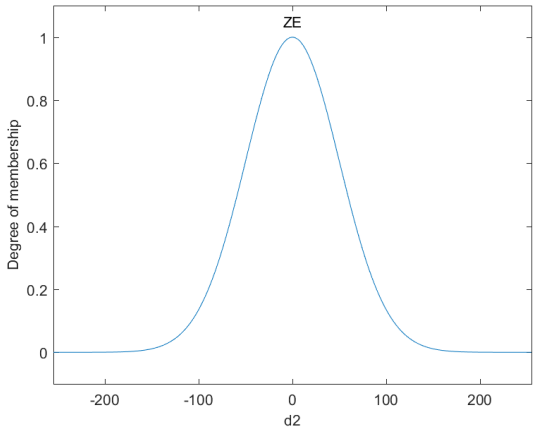

Figure 23: Input 1 Membership Function, Spread=50, d2

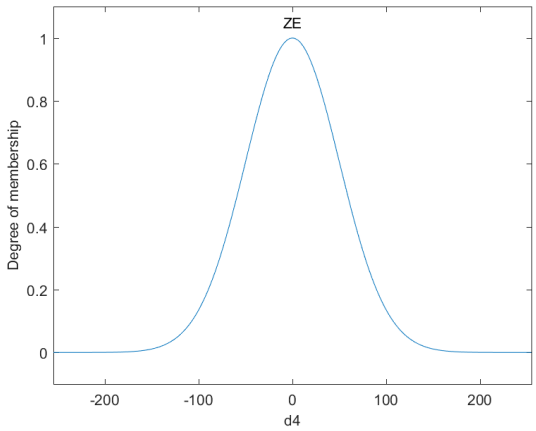

Figure 24: Input 2 Membership Function, Spread=50, d4

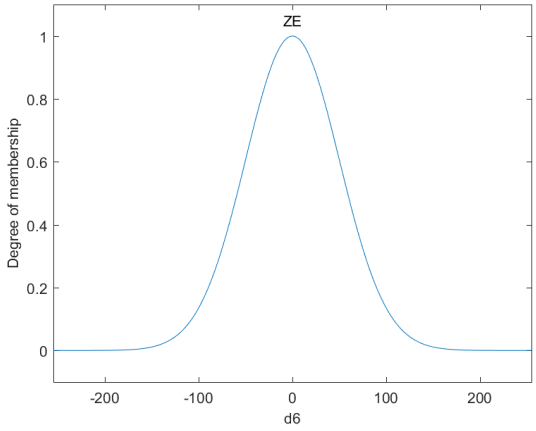

Figure 25: Input 3 Membership Function, Spread=50, d6

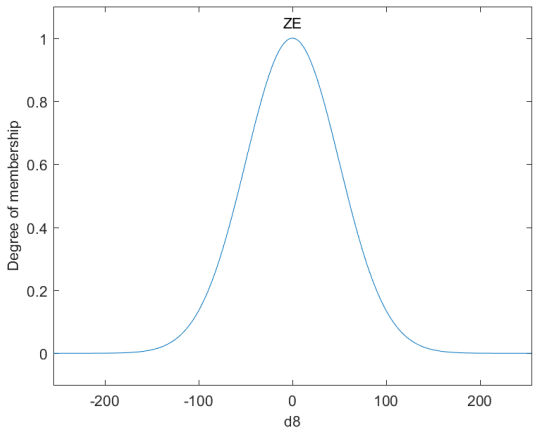

Figure 26: Input 4 Membership Function, Spread=50, d8

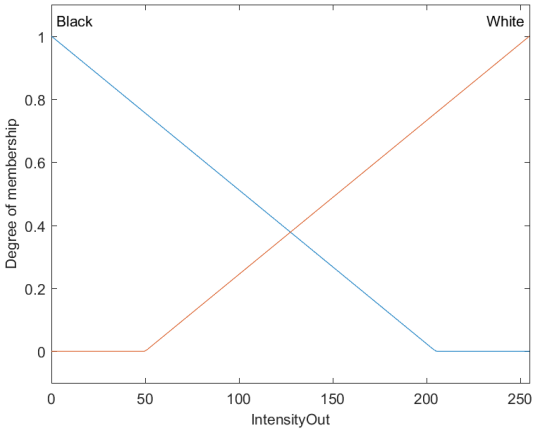

Figure 27: Output Membership Function, Spread=50

Figure 28: Processed Image With Scaling, Spread=50

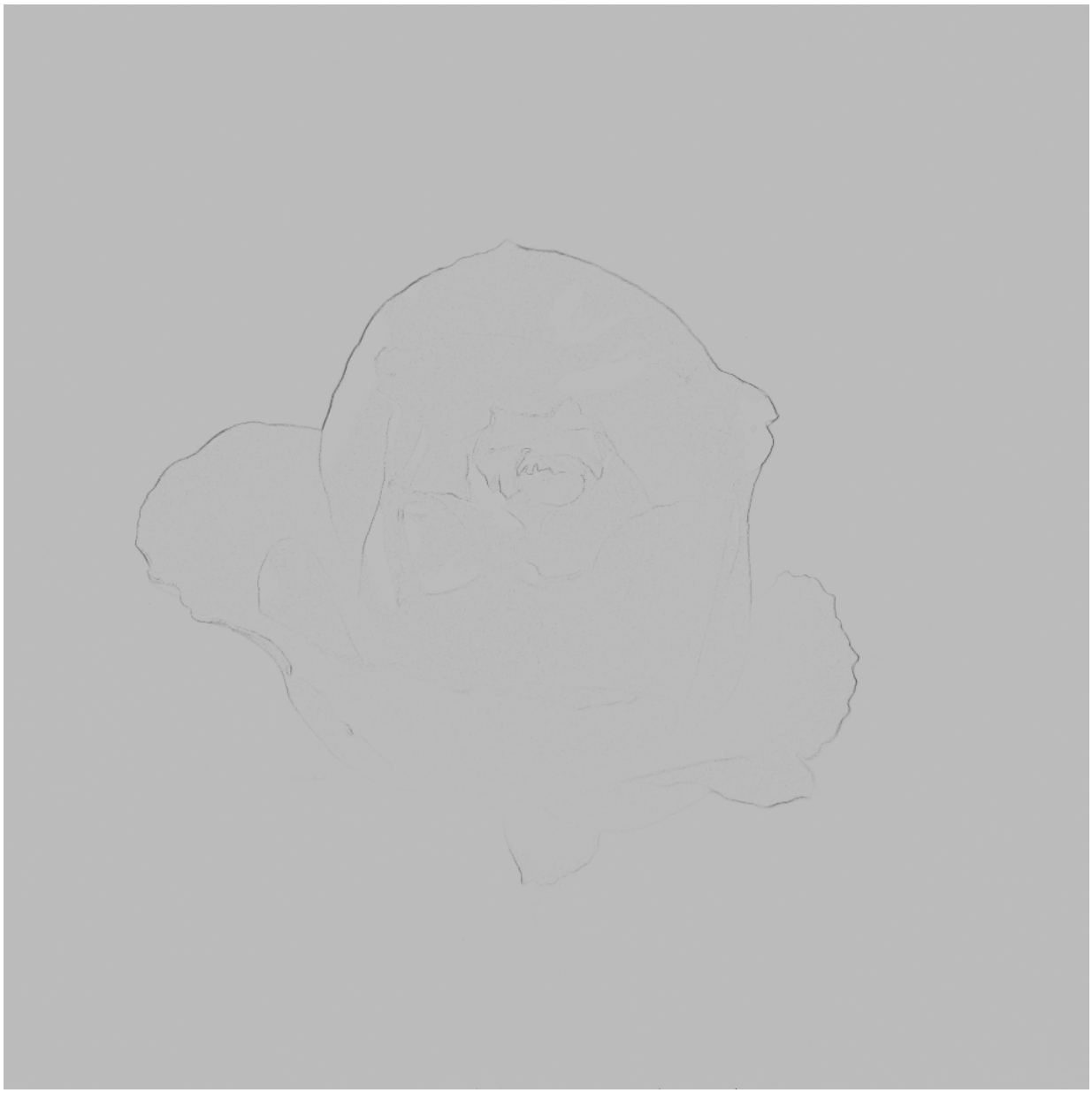

Figure 29: Processed Image Without Scaling, Spread=50

MATLAB Source Code

MAINSOURCECODEFILE

```
%
    ##########################################################################

%##                              Students names :
                                ##
%##                              Omar Rawashdeh
                                 ##
%##                               Harshal Raut
                                  ##
%##                               Utsav Shah
                                  ##
%
    ##########################################################################

%
    ##########################################################################

%##                          Digital Image Processing HW2
                          ##
%##                        Fuzzy Logic in processing Images
                        ##
%##                              Matlab R2016a was used
                           ##
%##                                  v9.0.0.341360
                            ##
%
    ##########################################################################


%prepare directories
warning('off', 'MATLAB:MKDIR:DirectoryExists');
mkdir('results');
mkdir(fullfile(pwd,'results'), 'Intensity Transformation'
    );
mkdir(fullfile(pwd,'results'), 'Spatial Filtering');
warning('on', 'MATLAB:MKDIR:DirectoryExists');

%constants for Intensity Transform
EDGES = 70;
DARK_LIMIT = 0+EDGES;
BRIGHT_LIMIT = 255-EDGES;
MID_GRAY = 127.5;
```

```
26
27 %% ========================================================================
28 %%                          Intensity Transform
29 %% ========================================================================
30 %create a new fuzzy inference system, for Intensity
       Transform
31 IntensityFuzzySystem = newfis('IntensityFIS', 'sugeno');
32 %define variables
33 IntensityFuzzySystem = addvar(IntensityFuzzySystem, '
       input', 'IntensityIn', [0 255]);
34 IntensityFuzzySystem = addvar(IntensityFuzzySystem, '
       output', 'IntensityOut', [0 255]);
35 %add mfs for the input
36 IntensityFuzzySystem = addmf(IntensityFuzzySystem, 'input
       ', 1, 'Dark', 'trapmf', [0 0 DARK_LIMIT MID_GRAY]);
37 IntensityFuzzySystem = addmf(IntensityFuzzySystem, 'input
       ', 1, 'Gray', 'trimf', [DARK_LIMIT MID_GRAY
       BRIGHT_LIMIT]);
38 IntensityFuzzySystem = addmf(IntensityFuzzySystem, 'input
       ', 1, 'Bright', 'trapmf', [MID_GRAY BRIGHT_LIMIT 255
       255]);
39 %add mfs for the output
40 IntensityFuzzySystem = addmf(IntensityFuzzySystem, '
       output', 1, 'Darker', 'constant', 0);
41 IntensityFuzzySystem = addmf(IntensityFuzzySystem, '
       output', 1, 'Gray', 'constant', 127);
42 IntensityFuzzySystem = addmf(IntensityFuzzySystem, '
       output', 1, 'Brighter', 'constant', 255);
43 %prepare rules list
44 rule1 = [1 1 1 1 ];
45 rule2 = [2 2 1 1 ];
46 rule3 = [3 3 1 1 ];
47 IntensityRulesList = [rule1;rule2;rule3];
48 %add the rules to the FIS
49 IntensityFuzzySystem = addrule(IntensityFuzzySystem,
       IntensityRulesList);
50 writefis(IntensityFuzzySystem, fullfile(pwd,'results', '
       Intensity Transformation', 'IntensityTransformFIS.fis'
       ));
51 %% ------------------------------------------------------------------------
```

```
52 %process an image using the FIS for Intesnsity
       Tranformation
53 Image1 = imread('Images for Project 3\Fig3-43a.jpg');
54 imwrite(Image1, fullfile(pwd,'results', 'Intensity
       Transformation', 'Used_Image.png'), 'png');
55 Histogram_Equalization_Image = histeq(Image1, 255);
56 imwrite(Histogram_Equalization_Image, fullfile(pwd,'
       results', 'Intensity Transformation', '
       Histogram_Equalization.png'), 'png');
57 Double_Image1_Input = mat2gray(Image1);
58 Double_Image1_Input = Double_Image1_Input*255;
59 Modified_Image1_Mat = zeros(size(Double_Image1_Input));
60 InputPixelsStream = reshape(Double_Image1_Input, size(
       Double_Image1_Input, 1)*size(Double_Image1_Input, 2),
       1);
61 OutputStream = evalfis(InputPixelsStream,
       IntensityFuzzySystem);
62 Modified_Image1_Mat = reshape(OutputStream, size(
       Modified_Image1_Mat, 1), size(Modified_Image1_Mat, 2))
       ;
63 Double_Image1_Output = mat2gray(Modified_Image1_Mat, [0
       255]);
64 Processed_Image1 = im2uint8(Double_Image1_Output);
65 imwrite(Processed_Image1, fullfile(pwd,'results', '
       Intensity Transformation', 'Transformed_Image.png'), '
       png');
66
67 Fig1 = figure('visible', 'off');
68 plotmf(IntensityFuzzySystem, 'input', 1);
69 print(Fig1, fullfile(pwd,'results', 'Intensity
       Transformation', 'Input1MFs.png'), '-dpng');
70
71
72
73
74 %constants for Spatial filtering
75 GAUSSIAN_SPREAD = 50;
76
77 %
       ==========================================================
78 %                        Spatial Filtering
79 %
       ==========================================================
```

```
80 %create a new fuzzy inference system, for Intensity
       Transform
81 SpatialFuzzySystem = newfis('SpatialFIS', 'mamdani');
82 %define variables
83 SpatialFuzzySystem = addvar(SpatialFuzzySystem, 'input',
       'd2', [-255 255]);
84 SpatialFuzzySystem = addvar(SpatialFuzzySystem, 'input',
       'd4', [-255 255]);
85 SpatialFuzzySystem = addvar(SpatialFuzzySystem, 'input',
       'd6', [-255 255]);
86 SpatialFuzzySystem = addvar(SpatialFuzzySystem, 'input',
       'd8', [-255 255]);
87 SpatialFuzzySystem = addvar(SpatialFuzzySystem, 'output',
        'IntensityOut', [0 255]);
88 %add mfs for the input
89 SpatialFuzzySystem = addmf(SpatialFuzzySystem, 'input',
       1, 'ZE', 'gaussmf', [GAUSSIAN_SPREAD 0]);
90 SpatialFuzzySystem = addmf(SpatialFuzzySystem, 'input',
       2, 'ZE', 'gaussmf', [GAUSSIAN_SPREAD 0]);
91 SpatialFuzzySystem = addmf(SpatialFuzzySystem, 'input',
       3, 'ZE', 'gaussmf', [GAUSSIAN_SPREAD 0]);
92 SpatialFuzzySystem = addmf(SpatialFuzzySystem, 'input',
       4, 'ZE', 'gaussmf', [GAUSSIAN_SPREAD 0]);
93 %add mfs for the output
94 SpatialFuzzySystem = addmf(SpatialFuzzySystem, 'output',
       1, 'Black', 'trimf', [0 0 205]);
95 SpatialFuzzySystem = addmf(SpatialFuzzySystem, 'output',
       1, 'White', 'trimf', [50 255 255]);
96 %prepare rules list
97 rule1 = [1 0 1 0 2 1 1];
98 rule2 = [0 0 1 1 2 1 1];
99 rule3 = [0 1 0 1 2 1 1];
100 rule4 = [1 1 0 0 2 1 1];
101 rule5 = [-1 0 0 -1 1 1 1];
102 rule6 = [0 -1 -1 0 1 1 1];
103 SpatialRulesList = [rule1;rule2;rule3;rule4;rule5;rule6];
104 %add the rules to the FIS
105 SpatialFuzzySystem = addrule(SpatialFuzzySystem,
       SpatialRulesList);
106 writefis(SpatialFuzzySystem, fullfile(pwd,'results', '
       Spatial Filtering', 'SpatialFilteringFIS.fis'));
107 %
```

```
108 %process an image using the FIS for spatial filtering
109 Image2 = imread('Images for Project 3\Fig2-19a.jpg');
```

```
% Image2 = imread('Untitled.png');
imwrite(Image2, fullfile(pwd,'results', 'Spatial
    Filtering', 'Used_Image.png'), 'png');
Double_Image2_Input = mat2gray(Image2);
Double_Image2_Input = Double_Image2_Input*255;

%prepare the input "differences stream"
InputDifferencesStream = zeros( (size(Double_Image2_Input
    , 1)-2) * (size(Double_Image2_Input, 2)-2) , 4);
for c=2:size(Double_Image2_Input, 2)-1
    C_Shift = (c-2)*(size(Double_Image2_Input, 1)-2);
    for r=2:size(Double_Image2_Input, 1)-1
        InputDifferencesStream(C_Shift+(r-1), 1) =
            Double_Image2_Input(r-1, c) -
            Double_Image2_Input(r, c);
        InputDifferencesStream(C_Shift+(r-1), 2) =
            Double_Image2_Input(r, c-1) -
            Double_Image2_Input(r, c);
        InputDifferencesStream(C_Shift+(r-1), 3) =
            Double_Image2_Input(r, c+1) -
            Double_Image2_Input(r, c);
        InputDifferencesStream(C_Shift+(r-1), 4) =
            Double_Image2_Input(r+1, c) -
            Double_Image2_Input(r, c);
    end
end

OutputStream2 = evalfis(InputDifferencesStream,
    SpatialFuzzySystem);
Modified_Image2_Mat = reshape(OutputStream2, size(
    Double_Image2_Input, 1)-2, size(Double_Image2_Input,
    2)-2);

Double_Image2_Output_Without_Scaling = mat2gray(
    Modified_Image2_Mat, [0 255]);
%scale using equations from textbook 2.6-10 and 2.6-11
Double_Image2_Output_With_Scaling =
    Double_Image2_Output_Without_Scaling - min(min(
    Double_Image2_Output_Without_Scaling));
Double_Image2_Output_With_Scaling =
    Double_Image2_Output_With_Scaling * 255 / max(max(
    Double_Image2_Output_With_Scaling));

Processed_Image2_Without_Scaling = im2uint8(
    Double_Image2_Output_Without_Scaling);
```

```
136 Processed_Image2_With_Scaling = im2uint8(
        Double_Image2_Output_With_Scaling);
137
138 imwrite(Processed_Image2_Without_Scaling, fullfile(pwd,'
        results', 'Spatial Filtering', 'Processed_Image
        without Scaling.png'), 'png');
139 imwrite(Processed_Image2_With_Scaling, fullfile(pwd,'
        results', 'Spatial Filtering', 'Processed_Image with
        Scaling.png'), 'png');
140
141 Fig2 = figure('visible', 'off');
142 plotmf(SpatialFuzzySystem, 'input', 1);
143 print(Fig2, fullfile(pwd,'results', 'Spatial Filtering',
        'Input1MFs.png'), '-dpng');
144 Fig3 = figure('visible', 'off');
145 plotmf(SpatialFuzzySystem, 'input', 2);
146 print(Fig3, fullfile(pwd,'results', 'Spatial Filtering',
        'Input2MFs.png'), '-dpng');
147 Fig4 = figure('visible', 'off');
148 plotmf(SpatialFuzzySystem, 'input', 3);
149 print(Fig4, fullfile(pwd,'results', 'Spatial Filtering',
        'Input3MFs.png'), '-dpng');
150 Fig5 = figure('visible', 'off');
151 plotmf(SpatialFuzzySystem, 'input', 4);
152 print(Fig5, fullfile(pwd,'results', 'Spatial Filtering',
        'Input4MFs.png'), '-dpng');
153 Fig6 = figure('visible', 'off');
154 plotmf(SpatialFuzzySystem, 'output', 1);
155 print(Fig6, fullfile(pwd,'results', 'Spatial Filtering',
        'Output1MFs.png'), '-dpng');
156 %showrule(IntensityFuzzySystem)
```

Bibliography

[1] Margaret Rouse. *What is Fuzzy Logic?* August 2016. URL: http://whatis.techtarget.com/definition/fuzzy-logic.

[2] *What is fuzzy logic*. Tech. rep. The MathWorks, Inc., 1994 - 2016. URL: https://www.mathworks.com/help/fuzzy/what-is-fuzzy-logic.html.