



**Politechnika
Śląska**

Przetwarzanie Obrazów Cyfrowych

**Raport z ćwiczenia nr. 4:
Klasyfikacja Obrazów**

Raport opracował:
Dawid Kania
Grupa 6 Semestr 7

Data wykonania ćwiczenia: 12.12.2022

Zadanie 2a

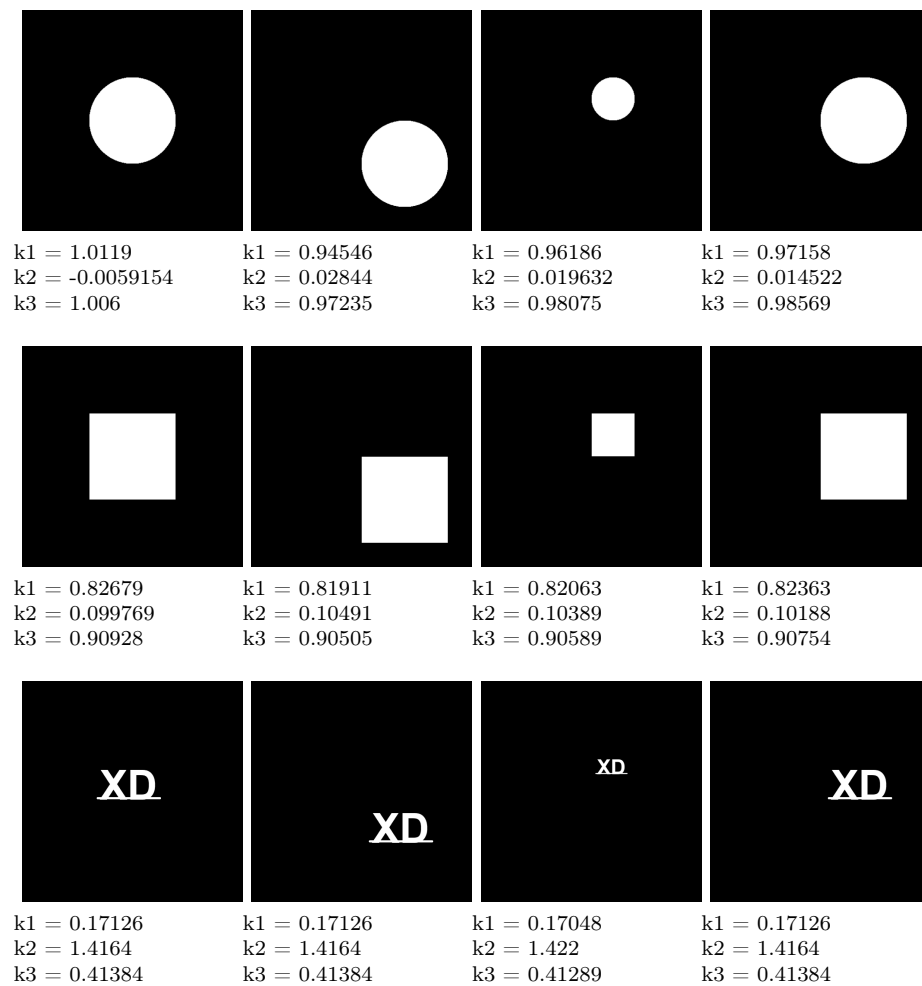
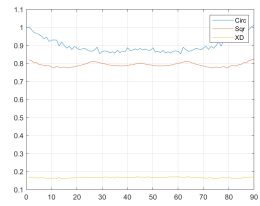
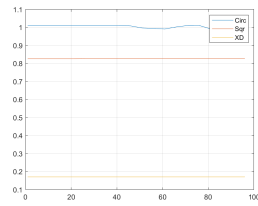


Figure 1: Porównanie

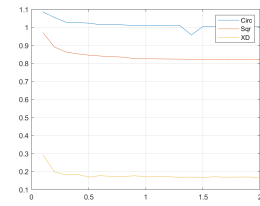
Zadanie 2b



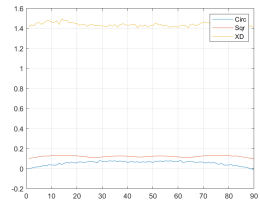
K1 - Rotacja



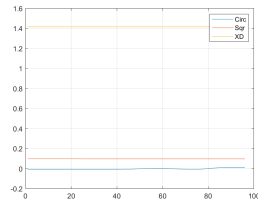
K1 - Skala



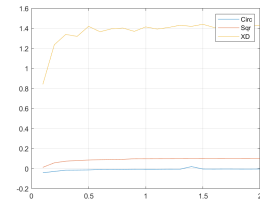
K1 - Translacja



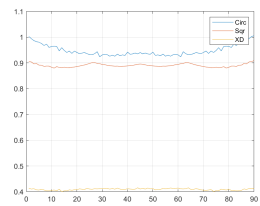
K2 - Rotacja



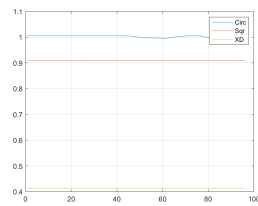
K2 - Skala



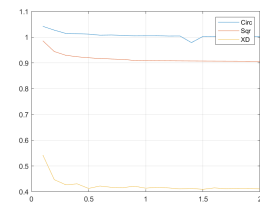
K2 - Translacja



K3 - Rotacja



K3 - Skala



K3 - Translacja

Figure 2: Porównanie

Zadanie 3

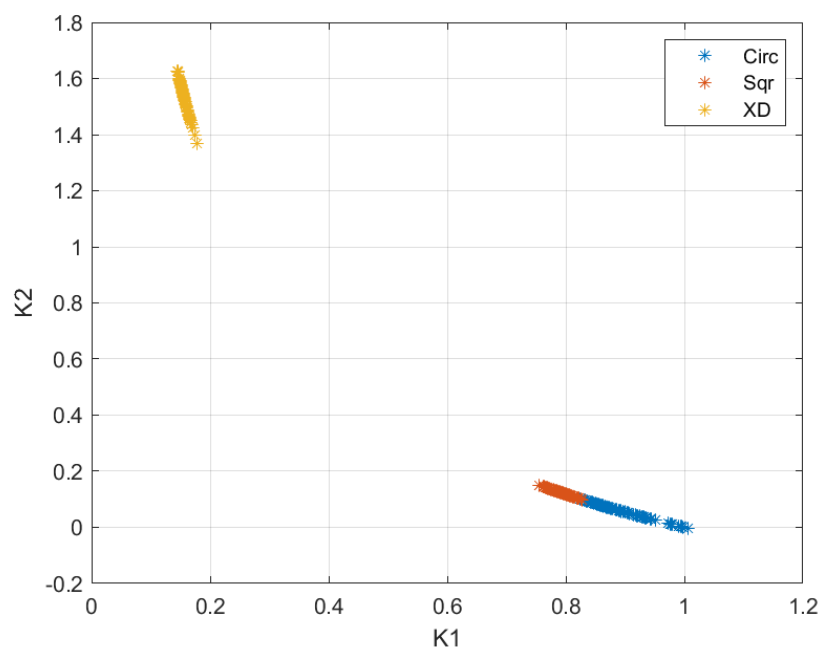


Figure 3: Porownanie

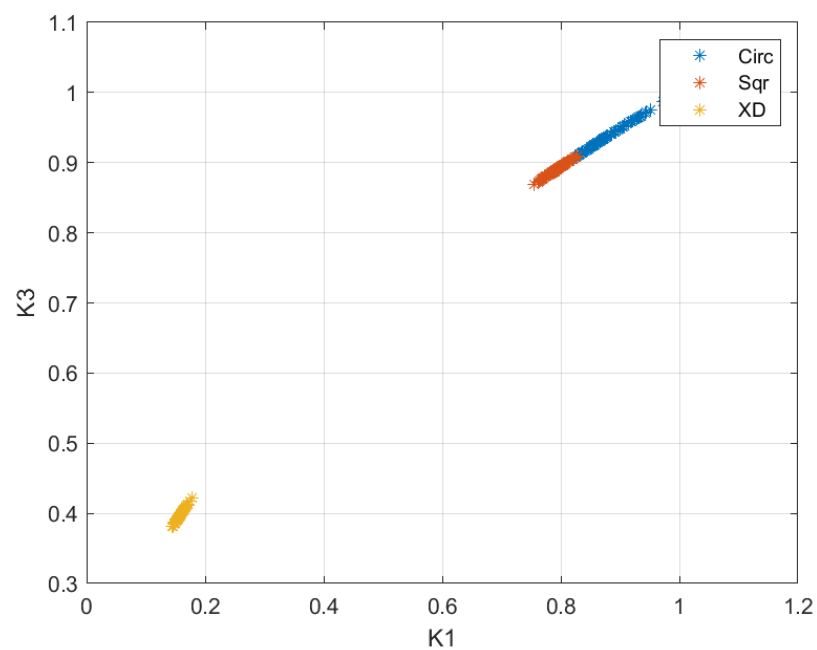


Figure 4: Porównanie

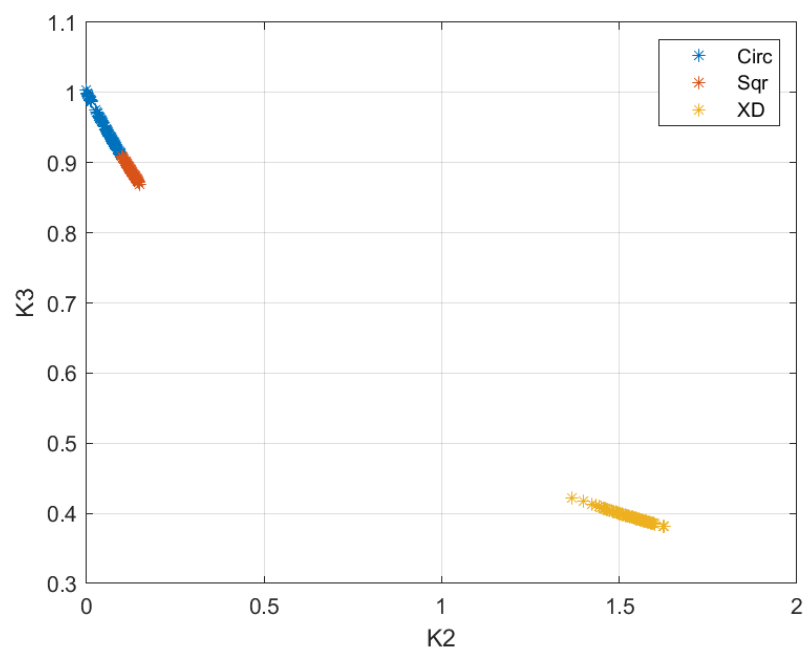


Figure 5: Porównanie

Kody programów

zad2a.m

```
1
2 clear all
3 close all
4 clc
5
6
7 figure
8
9 tiledlayout(3,4);
10
11
12 mkdir("../Zad2a/")
13 dest_folder = "../Zad2a";
14
15 K1 = zeros(12,1);
16 K2 = zeros(12,1);
17 K3 = zeros(12,1);
18
19 [K1( 1), K2( 1), K3( 1)] = show_img("../Zad2a/I1.png", @g_circles,
    [0 0], 0, [1 1]);
20 [K1( 2), K2( 2), K3( 2)] = show_img("../Zad2a/I2.png", @g_circles,
    [100 100], 0, [1 1]);
21 [K1( 3), K2( 3), K3( 3)] = show_img("../Zad2a/I3.png", @g_circles,
    [50 -50], 0, [.5 .5]);
22 [K1( 4), K2( 4), K3( 4)] = show_img("../Zad2a/I4.png", @g_circles,
    [100 0], 0, [1 1]);
23 [K1( 5), K2( 5), K3( 5)] = show_img("../Zad2a/I5.png", @g_squares,
    [0 0], 0, [1 1]);
24 [K1( 6), K2( 6), K3( 6)] = show_img("../Zad2a/I6.png", @g_squares,
    [100 100], 0, [1 1]);
25 [K1( 7), K2( 7), K3( 7)] = show_img("../Zad2a/I7.png", @g_squares,
    [50 -50], 0, [.5 .5]);
26 [K1( 8), K2( 8), K3( 8)] = show_img("../Zad2a/I8.png", @g_squares,
    [100 0], 0, [1 1]);
27 [K1( 9), K2( 9), K3( 9)] = show_img("../Zad2a/I9.png", @g_xd,
    [0 0], 0, [1 1]);
28 [K1(10), K2(10), K3(10)] = show_img("../Zad2a/I10.png", @g_xd,
    [100 100], 0, [1 1]);
29 [K1(11), K2(11), K3(11)] = show_img("../Zad2a/I11.png", @g_xd,
    [50 -50], 0, [.5 .5]);
30 [K1(12), K2(12), K3(12)] = show_img("../Zad2a/I12.png", @g_xd,
    [100 0], 0, [1 1]);
31
32
33 Latex = [
34     "\newcommand{\ww}{0.24} "
35     "\begin{figure}[H] "
36     "    \captionsetup[subfloat]{justification=raggedright,
37     singlelinecheck=false, position=bottom,labelformat=empty} % "
38     "    \subfloat[k1 = " + string(K1(1)) + " \k2 = " + string(K2
39     (1)) + " \k3 = " + string(K3(1)) + " ]{"
40     "    \includegraphics[width=\ww\linewidth]{ " + dest_folder
```

```

40 + "/I1.png}} \hfill% "
    \subfloat[k1 = " + string(K1(2)) + " \k2 = " + string(K2
41 (2)) + "\k3 = " + string(K3(2)) + " ]{"
    \includegraphics[width=\ww\linewidth]{ " + dest_folder
+ "/I2.png}} \hfill% "
42 \subfloat[k1 = " + string(K1(3)) + " \k2 = " + string(K2
(3)) + "\k3 = " + string(K3(3)) + " ]{"
43 \includegraphics[width=\ww\linewidth]{ " + dest_folder
+ "/I3.png}} \hfill%"
44 \subfloat[k1 = " + string(K1(4)) + " \k2 = " + string(K2
(4)) + "\k3 = " + string(K3(4)) + " ]{"
45 \includegraphics[width=\ww\linewidth]{ " + dest_folder
+ "/I4.png}} \hfill%"
46 ""
47 ""
48 \subfloat[k1 = " + string(K1(5)) + " \k2 = " + string(K2
(5)) + "\k3 = " + string(K3(5)) + " ]{"
49 \includegraphics[width=\ww\linewidth]{ " + dest_folder
+ "/I5.png}} \hfill% "
50 \subfloat[k1 = " + string(K1(6)) + " \k2 = " + string(K2
(6)) + "\k3 = " + string(K3(6)) + " ]{"
51 \includegraphics[width=\ww\linewidth]{ " + dest_folder
+ "/I6.png}} \hfill% "
52 \subfloat[k1 = " + string(K1(7)) + " \k2 = " + string(K2
(7)) + "\k3 = " + string(K3(7)) + " ]{"
53 \includegraphics[width=\ww\linewidth]{ " + dest_folder
+ "/I7.png}} \hfill%"
54 \subfloat[k1 = " + string(K1(8)) + " \k2 = " + string(K2
(8)) + "\k3 = " + string(K3(8)) + " ]{"
55 \includegraphics[width=\ww\linewidth]{ " + dest_folder
+ "/I8.png}} \hfill%"
56 ""
57 ""
58 \subfloat[k1 = " + string(K1(9)) + " \k2 = " + string(K2
(9)) + "\k3 = " + string(K3(9)) + " ]{"
59 \includegraphics[width=\ww\linewidth]{ " + dest_folder
+ "/I9.png}} \hfill% "
60 \subfloat[k1 = " + string(K1(10)) + " \k2 = " + string(K2
(10)) + "\k3 = " + string(K3(10)) + " ]{"
61 \includegraphics[width=\ww\linewidth]{ " + dest_folder
+ "/I10.png}} \hfill% "
62 \subfloat[k1 = " + string(K1(11)) + " \k2 = " + string(K2
(11)) + "\k3 = " + string(K3(11)) + " ]{"
63 \includegraphics[width=\ww\linewidth]{ " + dest_folder
+ "/I11.png}} \hfill%"
64 \subfloat[k1 = " + string(K1(12)) + " \k2 = " + string(K2
(12)) + "\k3 = " + string(K3(12)) + " ]{"
65 \includegraphics[width=\ww\linewidth]{ " + dest_folder
+ "/I12.png}} \hfill%"
66 ""
67 \caption{Porownanie}
68 ""
69 \end{figure} "
70 \let\ww\undefined "
71 ];
72
73 Latex = join(Latex,[''],2);

```



```

74 Latex = join(Latex,[newline],1);
75
76 fid = fopen(dest_folder + "/result.tex",'wt');
77 fprintf(fid,"%s", Latex);
78 fclose(fid);
79
80
81
82
83
84
85 function [k1, k2, k3] = show_img(dest, func, trans, rot, scale)
86     nexttile
87     I1 = func(trans,rot,scale);
88     imshow(I1);
89     imwrite(I1, dest);
90     k1 = wspolczynnik_kompaktowosci(I1);
91     k2 = wspolczynnik_malinowskiej(I1);
92     k3 = wspolczynnik_Mz(I1);
93     title("K1 = " + string(k1) + " K2 = " + string(k2) + " K3 = " +
94           string(k3));
95 end

```

zad2b.m

```

1 clear all
2 close all
3 clc
4
5
6 mkdir("../Zad2b")
7
8 %%
9
10 [k1_circ, k2_circ, k3_circ, k1_sqr, k2_sqr, k3_sqr, k1_xd, k2_xd,
    k3_xd] = get_coefficients_rot()
11
12 plot_and_save("../Zad2b/k1_rot.png", [1:1:90], k1_circ, k1_sqr,
    k1_xd);
13 plot_and_save("../Zad2b/k2_rot.png", [1:1:90], k2_circ, k2_sqr,
    k2_xd);
14 plot_and_save("../Zad2b/k3_rot.png", [1:1:90], k3_circ, k3_sqr,
    k3_xd);
15
16 %%
17
18 [k1_circ, k2_circ, k3_circ, k1_sqr, k2_sqr, k3_sqr, k1_xd, k2_xd,
    k3_xd] = get_coefficients_scale()
19
20 plot_and_save("../Zad2b/k1_scale.png", [1:1:100], k1_circ, k1_sqr,
    k1_xd);
21 plot_and_save("../Zad2b/k2_scale.png", [1:1:100], k2_circ, k2_sqr,
    k2_xd);
22 plot_and_save("../Zad2b/k3_scale.png", [1:1:100], k3_circ, k3_sqr,
    k3_xd);
23
24 %%
25
26 [k1_circ, k2_circ, k3_circ, k1_sqr, k2_sqr, k3_sqr, k1_xd, k2_xd,
    k3_xd] = get_coefficients_trans()
27
28 plot_and_save("../Zad2b/k1_trans.png", [1:5:100], k1_circ, k1_sqr,
    k1_xd);
29 plot_and_save("../Zad2b/k2_trans.png", [1:5:100], k2_circ, k2_sqr,
    k2_xd);
30 plot_and_save("../Zad2b/k3_trans.png", [1:5:100], k3_circ, k3_sqr,
    k3_xd);
31
32
33 %%
34
35 dest_folder = "../Zad2b"
36
37 Latex = [
38     "\newcommand{\ww}{0.32} "
39     "\begin{figure}[H] "
40     "    \captionsetup[subfloat]{justification=raggedright,
41         singlelinecheck=false, position=bottom,labelformat=empty} % "
42     ""
43     "    \subfloat[K1 - Rotacja]{"
```

```

43     "        \includegraphics[width=\ww\linewidth]{ " + dest_folder
44 + "/k1_rot.png}} \hfill% "
45     "        \subfloat[K1 - Skala]{ "
46 + "/k1_trans.png}} \hfill% "
47     "        \includegraphics[width=\ww\linewidth]{ " + dest_folder
48 + "/k1_scale.png}} \hfill% "
49     ""
50     "        \subfloat[K2 - Rotacja]{ "
51 + "/k2_rot.png}} \hfill% "
52     "        \subfloat[K2 - Skala]{ "
53 + "/k2_trans.png}} \hfill% "
54     "        \subfloat[K2 - Translacja]{ "
55 + "/k2_scale.png}} \hfill% "
56     ""
57     "        \subfloat[K3 - Rotacja]{ "
58 + "/k3_rot.png}} \hfill% "
59     "        \subfloat[K3 - Skala]{ "
60 + "/k3_trans.png}} \hfill% "
61     "        \subfloat[K3 - Translacja]{ "
62 + "/k3_scale.png}} \hfill% "
63     ""
64     ""
65     "\caption{Porownanie}"
66     ""
67     "\end{figure} "
68     "\let\ww\undefined "
69 ];
70
71 Latex = join(Latex,[''],2);
72 Latex = join(Latex,[newline],1);
73
74 fid = fopen(dest_folder + "/result.tex",'wt');
75 fprintf(fid,"%s", Latex);
76 fclose(fid);
77
78
79
80 %%
81
82
83 function plot_and_save(dest,x,circ, sqr, xd)
84 % K1 - rot
85 hold off;
86 plot(x, circ); hold on;
87 plot(x, sqr);
88 plot(x, xd);
89 grid on;
90 legend("Circ", "Sqr", "XD");

```

```

91     saveas(gcf, dest);
92
93 end
94
95
96
97 function [k1_circ, k2_circ, k3_circ, k1_sqr, k2_sqr, k3_sqr, k1_xd,
    k2_xd, k3_xd] = get_coeficients_rot()
98
99     translation = [0 0];
100     scale = [1 1];
101     all_rotations = [1:1:90];
102
103     i = 1;
104     for rotation = all_rotations;
105
106         disp(rotation);
107
108         I1 = g_circles(translation, rotation, scale );
109         [k1_circ(i), k2_circ(i), k3_circ(i)] = get_coeficients(I1);
110
111         I1 = g_squares(translation, rotation, scale );
112         [k1_sqr(i), k2_sqr(i), k3_sqr(i)] = get_coeficients(I1);
113
114         I1 = g_xd(translation, rotation, scale );
115         [k1_xd(i), k2_xd(i), k3_xd(i)] = get_coeficients(I1);
116
117         i = i + 1;
118     end
119 end
120
121
122
123 function [k1_circ, k2_circ, k3_circ, k1_sqr, k2_sqr, k3_sqr, k1_xd,
    k2_xd, k3_xd] = get_coeficients_scale()
124
125     translation = [0 0];
126     all_scale = [.1:.1:2];
127     rotation = 0;
128
129     i = 1;
130     for s1 = all_scale;
131
132         scale = [s1, s1]
133
134         disp(s1);
135
136         I1 = g_circles(translation, rotation, scale );
137         [k1_circ(i), k2_circ(i), k3_circ(i)] = get_coeficients(I1);
138
139         I1 = g_squares(translation, rotation, scale );
140         [k1_sqr(i), k2_sqr(i), k3_sqr(i)] = get_coeficients(I1);
141
142         I1 = g_xd(translation, rotation, scale );
143         [k1_xd(i), k2_xd(i), k3_xd(i)] = get_coeficients(I1);
144
145         i = i + 1;

```

```

146     end
147
148 end
149
150 function [k1_circ, k2_circ, k3_circ, k1_sqr, k2_sqr, k3_sqr, k1_xd,
151          k2_xd, k3_xd] = get_coeficients_trans()
152
153     all_translation = [1:5:100];
154     scale = [1 1];
155     rotation = 0;
156
157     i = 1;
158     for t1 = all_translation;
159
160         translation = [t1 0];
161
162         disp(t1);
163
164         I1 = g_circles(translation, rotation, scale );
165         [k1_circ(i), k2_circ(i), k3_circ(i)] = get_coeficients(I1);
166
167         I1 = g_squares(translation, rotation, scale );
168         [k1_sqr(i), k2_sqr(i), k3_sqr(i)] = get_coeficients(I1);
169
170         I1 = g_xd(translation, rotation, scale );
171         [k1_xd(i), k2_xd(i), k3_xd(i)] = get_coeficients(I1);
172
173         i = i + 1;
174     end
175 end
176
177
178 function [k1, k2, k3] = get_coeficients(I1)
179
180
181     k1 = wspolczynnik_kompaktowosci(I1);
182     k2 = wspolczynnik_malinowskiej(I1);
183     k3 = wspolczynnik_Mz(I1);
184
185 end

```

zad3.m

```
1
2 clear all
3 close all
4 clc
5
6
7
8 circ_source = "../images_gen/circ" + string(1:100) + ".png";
9 sqr_source = "../images_gen/sqr" + string(1:100) + ".png";
10 xD_source = "../images_gen/xD" + string(1:100) + ".png";
11
12
13
14
15 % plot3_coeficients(circ_source); hold on;
16 % plot3_coeficients(sqr_source);
17 % plot3_coeficients(xD_source);
18
19 mkdir("../Zad3")
20 plot_coeficients(circ_source, sqr_source, xD_source)
21
22
23
24 legend("Circ", "Sqr", "XD");
25 grid on;
26
27
28 function plot_coeficients(src_vec1, src_vec2, src_vec3)
29     [k1_v1,k2_v1,k3_v1] = get_coeficient_vectors(src_vec1);
30     [k1_v2,k2_v2,k3_v2] = get_coeficient_vectors(src_vec2);
31     [k1_v3,k2_v3,k3_v3] = get_coeficient_vectors(src_vec3);
32
33     hold off;
34     plot(k1_v1, k2_v1, '*'); hold on;
35     plot(k1_v2, k2_v2, '*');
36     plot(k1_v3, k2_v3, '*');
37     grid on;
38     legend("Circ", "Sqr", "XD");
39     xlabel("K1");
40     ylabel("K2");
41     saveas(gcf, "../Zad3/k1k2.png");
42
43
44     hold off;
45     plot(k1_v1, k3_v1, '*'); hold on;
46     plot(k1_v2, k3_v2, '*');
47     plot(k1_v3, k3_v3, '*');
48     grid on;
49     legend("Circ", "Sqr", "XD");
50     xlabel("K1");
51     ylabel("K3");
52     saveas(gcf, "../Zad3/k1k3.png");
53
54
55     hold off;
```

```

56     plot(k2_v1, k3_v1, '*'); hold on;
57     plot(k2_v2, k3_v2, '*');
58     plot(k2_v3, k3_v3, '*');
59     grid on;
60     legend("Circ", "Sqr", "XD");
61     xlabel("K2");
62     ylabel("K3");
63     saveas(gcf, "../Zad3/k2k3.png");
64
65 end
66
67 function plot3_coefficients(src_vec)
68     [k1,k2,k3] = get_coefficient_vectors(src_vec);
69     plot3(k1,k2,k3, "*");
70     xlabel("K1");
71     ylabel("K2");
72     zlabel("K3");
73 end
74
75
76
77 function [k1, k2, k3] = get_coefficient_vectors(src_vec)
78
79     i = 1;
80
81     for source = src_vec
82
83         disp(source);
84         I1 = imread(source);
85         I1 = double(I1) ./ 255;
86
87         if(size(I1,3) ~= 1)
88             I1 = rgb2gray(I1);
89         end
90
91         [k1(i), k2(i), k3(i)] = get_coefficients(I1);
92
93         i = i + 1;
94     end
95
96 end
97
98
99
100
101 function [k1, k2, k3] = get_coefficients(I1)
102
103     k1 = wspolczynnik_kompaktowosci(I1);
104     k2 = wspolczynnik_malinowskiej(I1);
105     k3 = wspolczynnik_Mz(I1);
106
107 end

```

wspolczynnik_kompaktowosci.m

```
1 function K = wspolczynnik_kompaktowosci(I1)
2
3     I1 = I1 > 0.5;
4
5     all1 = regionprops(I1, "all");
6
7     S = all1.Area;
8     L = all1.Perimeter;
9
10
11     K = 4*pi*S/L^2;
12
13 end
```


wspolczynnik_malinowskiej.m

```
1 function M = wspolczynnik_malinowskiej(I1)
2
3     I1 = I1 > 0.5;
4
5     all1 = regionprops(I1, "all");
6
7     S = all1.Area;
8     L = all1.Perimeter;
9
10
11
12     M = L / (2*sqrt(pi*S)) - 1;
13
14 end
```

wspolczynnik_Mz.m

```
1 function Mz = wspolczynnik_Mz(I1)
2
3     I1 = I1 > 0.5;
4
5     all1 = regionprops(I1, "all");
6
7     S = all1.Area;
8     L = all1.Perimeter;
9
10
11     Mz = 2 * sqrt(pi*S) / L;
12
13 end
```

g_circles.m

```
1
2 function I1 = g_circles(translation, rotation, scale)
3
4     % translation = [0 0];
5     % rotation = [0];
6     % scale = [0 0];
7
8
9     canvas_size = [512, 512];
10    radius = 100;
11    center = canvas_size./2;
12
13    % xy coordinates
14    [x,y] = meshgrid(1:canvas_size(1), 1:canvas_size(2));
15    x = x - center(1);
16    y = y - center(2);
17
18    % apply scale
19    x = x ./ scale(1);
20    y = y ./ scale(2);
21
22
23
24    % draw circle
25    I1 = (x.^2 + y.^2) < radius.^2;
26    I1 = double(I1);
27
28    I1 = imtranslate(I1, translation);
29    I1 = imrotate(I1, rotation);
30
31
32    %I1 = [ zeros(size(I1,1), 500), I1, zeros(size(I1,1), 500)];
33    %I1 = [ zeros(500, size(I1,2)); I1; zeros(500, size(I1,2))];
34    %cutted_part = ((scale-1).*canvas_size)/2 + 500;
35    cutted_part = [(size(I1,1) - 512), (size(I1,2) - 512)]/2;
36    crop_rect = [cutted_part, 511, 511];
37    I1 = imcrop(I1, crop_rect);
38
39
40    I1 = imresize(I1, [512, 512]);
41
42 end
```

g_squares.m

```
1
2 function I1 = g_squares(translation, rotation, scale)
3
4     % translation = [0 0];
5     % rotation = [0];
6     % scale = [0 0];
7
8
9     canvas_size = [512, 512];
10    size1 = 100;
11    center = canvas_size./2;
12
13    % xy coordinates
14    [x,y] = meshgrid(1:canvas_size(1), 1:canvas_size(2));
15    x = x - center(1);
16    y = y - center(2);
17
18    % apply scale
19    x = x ./ scale(1);
20    y = y ./ scale(2);
21
22
23
24    % draw square
25    I1 = (abs(x) < size1) & (abs(y) < size1);
26    I1 = double(I1);
27
28    I1 = imtranslate(I1, translation);
29    I1 = imrotate(I1, rotation);
30
31
32    cutted_part = [(size(I1,1) - 512), (size(I1,2) - 512)]/2;
33    crop_rect = [cutted_part, 511, 511];
34    I1 = imcrop(I1, crop_rect);
35
36
37
38    I1 = imresize(I1, [512, 512]);
39
40 end
```

g_xd.m

```
1
2 function I1 = g_xd(translation, rotation, scale)
3
4     source = "../images/xD.png";
5
6     % translation = [0 0];
7     % rotation = [0];
8     % scale = [0 0];
9
10    I1 = imread(source);
11    I1 = double(I1) ./ 255;
12
13    if(size(I1,3) ~= 1)
14        I1 = rgb2gray(I1);
15    end
16
17    canvas_size = [512, 512];
18    size1 = 100;
19    center = canvas_size./2;
20
21
22    I1 = imrotate(I1, rotation);
23
24
25    I1 = imresize(I1, scale.*canvas_size);
26
27
28
29    I1 = [ zeros(size(I1,1), 500), I1, zeros(size(I1,1), 500)];
30    I1 = [ zeros(500, size(I1,2)); I1; zeros(500, size(I1,2))];
31
32    cutted_part = ((scale-1).*canvas_size)/2 + 500;
33
34    crop_rect = [cutted_part, 511, 511];
35    I1 = imcrop(I1, crop_rect);
36
37
38    I1 = imtranslate(I1, translation);
39
40    I1 = imresize(I1, [512, 512]);
41
42 end
```

save_all.m

```
1 clear all
2 close all
3 clc
4
5
6 save_circles
7 save_squares
8 save_xd
```

save_circles.m

```
1 clear all
2 close all
3 clc
4
5
6
7
8 destination = "../images_gen";
9 mkdir(destination);
10
11 for i = 1:100;
12
13     translation = (rand(1,2) - .5) * 200
14     rotation = rand(1) * 360
15     scale = rand(1,2) * .3 + .5
16
17     I1 = g_circles(translation, rotation, scale);
18
19     filename = destination + "/circ" + string(i) + ".png";
20     imwrite(I1, filename);
21
22 end
```

save_squares.m

```
1 clear all
2 close all
3 clc
4
5
6 destination = "../images_gen";
7 mkdir(destination);
8
9
10 for i = 1:100;
11
12     translation = (rand(1,2) - .5) * 200
13     rotation = rand(1) * 360
14     scale = rand(1,2) * .3 + .5
15
16     I1 = g_squares(translation, rotation, scale);
17
18     filename = destination + "/sqr" + string(i) + ".png";
19     imwrite(I1, filename);
20
21 end
```


save_xd.m

```
1 clear all
2 close all
3 clc
4
5
6
7 destination = "../images_gen";
8 mkdir(destination);
9
10 for i = 1:100;
11
12     translation = (rand(1,2) - .5) * 100
13     rotation = rand(1) * 360
14     scale = rand(1,2) * .5 + 2
15
16     I1 = g_xd(translation, rotation, scale);
17
18
19     filename = destination + "/xD" + string(i) + ".png";
20     imwrite(I1, filename);
21
22 end
```