

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

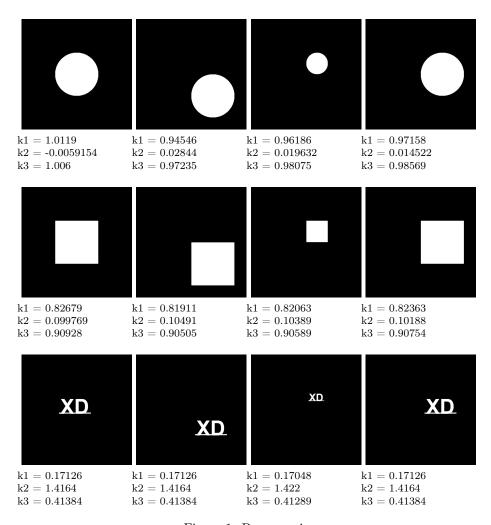


Figure 1: Porownanie

Zadanie 2b

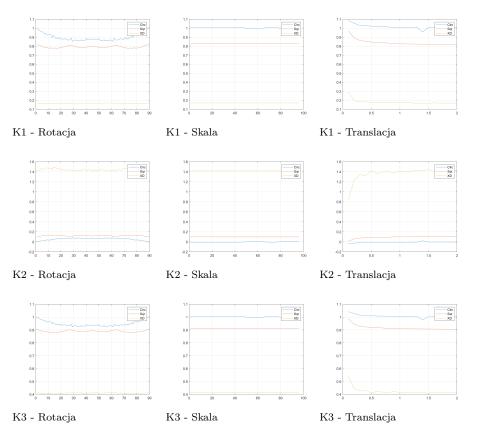


Figure 2: Porownanie

Zadanie 3

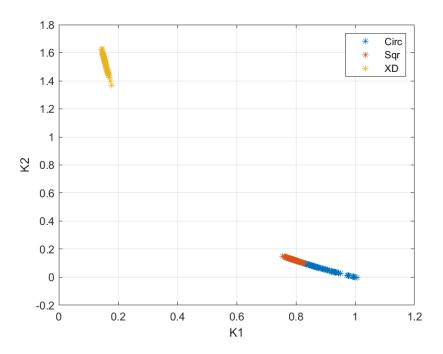


Figure 3: Porownanie

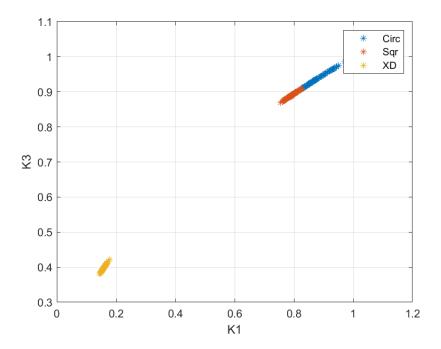


Figure 4: Porownanie

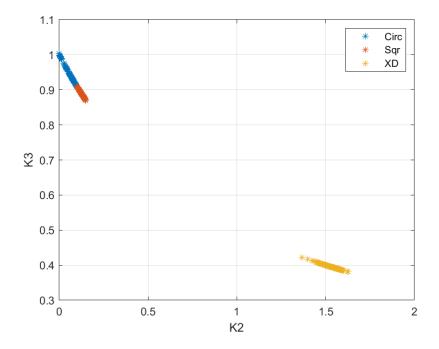


Figure 5: Porownanie

Kody programów

zad2a.m

```
2 clear all
3 close all
4 clc
7 figure
9 tiledlayout(3,4);
1.1
12 mkdir("../Zad2a/")
dest_folder = "../Zad2a";
14
15 K1 = zeros(12,1);
16 \text{ K2} = zeros(12,1);
17 \text{ K3} = zeros(12,1);
18
19 [K1(1), K2(1), K3(1)] = show_img("../Zad2a/I1.png", @g_circles,
               0, [1 1]);
      [0 0],
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]);
  [K1(11), K2(11), K3(11)] = show_img("../Zad2a/I11.png", @g_xd,
       [50 -50], 0, [.5 .5]);
  [K1(12), K2(12), K3(12)] = show_img("../Zad2a/I12.png", @g_xd,
       [100 0], 0, [1 1]);
31
32
33 Latex = [
      "\newcommand{\ww}\{0.24\}"
      "\begin{figure}[H] "
35
          \captionsetup[subfloat]{justification=raggedright,
36
      \verb|singlelinecheck=false|, position=bottom|, labelformat=empty| % "
37
           (1)) + "\\k3 = " + string(K3(1)) + " ]{"
              \includegraphics[width=\ww\linewidth]{" + dest_folder
```

```
+ "/I1.png}} \hfill% "
           \ \left[k1 = " + string(K1(2)) + " \setminus k2 = " + string(K2)\right]
      (2)) + "\\k3 = " + string(K3(2)) + " ]{"
               \includegraphics[width=\ww\linewidth]{" + dest_folder
41
      + "/I2.png}} \hfill% "
           \left[k1 = " + string(K1(3)) + " \right] = " + string(K2)
42
      (3)) + "\\k3 = " + string(K3(3)) + " ]{"
               \includegraphics[width=\ww\linewidth]{" + dest_folder
43
      + "/I3.png}} \hfill%"
           \sl k1 = " + string(K1(4)) + " \setminus k2 = " + string(K2
44
      (4)) + " \setminus k3 = " + string(K3(4)) + " ]{"}
              \includegraphics[width=\ww\linewidth]{" + dest_folder
      + "/I4.png}} \hfill%"
      0,0
47
           \left[k1 = " + string(K1(5)) + " \right] = " + string(K2)
48
      (5)) + "\\k3 = " + string(K3(5)) + " ]{"
               \includegraphics[width=\ww\linewidth]{" + dest_folder
49
      + "/I5.png}} \hfill% "
           \left[k1 = " + string(K1(6)) + " \setminus k2 = " + string(K2)\right]
      (6)) + "\k3 = " + string(\k3(6)) + " ]{"
               \includegraphics[width=\ww\linewidth]{" + dest_folder
      + "/I6.png}} \hfill% "
           \sl k1 = " + string(K1(7)) + " \setminus k2 = " + string(K2)
      (7)) + "\\k3 = " + string(K3(\overline{7})) + " ]{"
               \includegraphics[width=\ww\linewidth]{" + dest_folder
      + "/I7.png}} \hfill%"
           \left[k1 = " + string(K1(8)) + " \right] 
54
      (8)) + "\\k3 = " + string(K3(8)) + " ]{"
               \includegraphics[width=\ww\linewidth]{" + dest_folder
      + "/I8.png}} \hfill%"
56
      0.0
           \left[k1 = " + string(K1(9)) + " \setminus k2 = " + string(K2)\right]
58
      (9)) + "\\k3 = " + string(K3(9)) + " ]{"
               \includegraphics[width=\ww\linewidth]{" + dest_folder
      + "/I9.png}} \hfill% "
           \left[k1 = " + string(K1(10)) + " \right] 
      (10)) + "\\k3 = " + string(K3(10)) + " ]{"
               \includegraphics[width=\ww\linewidth]{" + dest_folder
61
      + "/I10.png}} \hfill% '
           \left[k1 = " + string(K1(11)) + " \right] 
62
      (11)) + "\\k3 = " + string(K3(11)) + " ]{"
               \includegraphics[width=\ww\linewidth]{" + dest_folder
63
      + "/I11.png}} \hfill%"
           \subfloat[k1 = " + string(K1(12)) + " \\k2 = " + string(K2
      (12)) + "\\k3 = " + string(K3(12)) + " ]{"
               \includegraphics[width=\ww\linewidth]{" + dest_folder
      + "/I12.png}} \hfill%"
66
      "\caption{Porownanie}"
67
68
      "\end{figure} "
69
      "\let\ww\undefined "
70
71
          ];
73 Latex = join(Latex,[''],2);
```

```
74 Latex = join(Latex,[newline],1);
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)
      nexttile
86
       I1 = func(trans, rot, scale);
87
       imshow(I1);
88
89
       imwrite(I1, dest);
       k1 = wspolczynnik_kompaktowosci(I1);
90
       k2 = wspolczynnik_malinowskiej(I1);
91
      k3 = wspolczynnik_Mz(I1);
title("K1 = " + string(k1) + " K2 = " + string(k2) + " K3 = " +
92
93
        string(k3));
94
95 end
```

zad2b.m

```
1 clear all
2 close all
3 clc
6 mkdir("../Zad2b")
8 %%
10 [k1_circ, k2_circ, k3_circ, k1_sqr, k2_sqr, k3_sqr, k1_xd, k2_xd,
      k3_xd] = get_coeficients_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
  [k1_circ, k2_circ, k3_circ, k1_sqr, k2_sqr, k3_sqr, k1_xd, k2_xd,
      k3_xd] = get_coeficients_scale()
plot_and_save("../Zad2b/k1_scale.png", [.1:.1:2] ,k1_circ, k1_sqr,
      k1_xd);
21 plot_and_save("../Zad2b/k2_scale.png", [.1:.1:2] ,k2_circ, k2_sqr,
      k2_xd);
22 plot_and_save("../Zad2b/k3_scale.png", [.1:.1:2] ,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_coeficients_trans()
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 = [
      "\newcommand{\ww}\{0.32\}"
38
       "\begin{figure}[H] "
39
      " \captionsetup[subfloat]{justification=raggedright,
40
      singlelinecheck=false, position=bottom,labelformat=empty} % "
41
           \subfloat[K1 - Rotacja]{"
42
```

```
\includegraphics[width=\ww\linewidth]{" + dest_folder
43
       + "/k1_rot.png}} \hfill%
            \subfloat[K1 - Skala]{"
44
                \includegraphics[width=\ww\linewidth]{" + dest_folder
45
       + "/k1_trans.png}} \hfill% "
            \subfloat[K1 - Translacja]{"
46
                 \includegraphics[width=\ww\linewidth]{" + dest_folder
47
       + "/k1_scale.png}}
                            \hfill% "
            \subfloat[K2 - Rotacja]{"
49
                 \includegraphics[width=\ww\linewidth]{" + dest_folder
50
       + "/k2_rot.png}} \hfill% "
            \subfloat[K2 - Skala]{"
                \includegraphics[width=\ww\linewidth]{" + dest_folder
      + "/k2_trans.png}} \hfill% "
" \subfloat[K2 - Translacja]{"
53
54
                \includegraphics[width=\ww\linewidth]{" + dest_folder
       + "/k2_scale.png}}
                           \hfill% "
55
            \subfloat[K3 - Rotacja]{"
56
                 \includegraphics[width=\ww\linewidth]{" + dest_folder
       + "/k3_rot.png}} \hfill%"
            \subfloat[K3 - Skala]{"
58
                \includegraphics[width=\ww\linewidth]{" + dest_folder
59
       + "/k3_trans.png}} \hfill%"
" \subfloat[K3 - Translacja]{"
                \includegraphics[width=\ww\linewidth]{" + dest_folder
61
       + "/k3_scale.png}} \hfill%"
62
       0.0
63
64
       "\caption{Porownanie}"
65
66
       "\end{figure} "
67
       "\let\ww\undefined "
68
69
          ];
70
71 Latex = join(Latex,[''],2);
72 Latex = join(Latex,[newline],1);
74 fid = fopen(dest_folder + "/result.tex",'wt');
75 fprintf(fid,"%s", Latex);
76 fclose(fid);
77
79
80
81 %%
82
  function plot_and_save(dest,x,circ, sqr, xd)
83
       % K1 - rot
84
       hold off;
85
       plot(x, circ); hold on;
86
      plot(x, sqr);
plot(x, xd);
87
88
       grid on;
89
90 legend("Circ", "Sqr", "XD");
```

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

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

zad3.m

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

```
plot(k2_v1, k3_v1, '*'); hold on;
plot(k2_v2, k3_v2, '*');
plot(k2_v3, k3_v3, '*');
56
57
58
59
        grid on;
        legend("Circ", "Sqr", "XD");
60
        xlabel("K2");
61
        ylabel("K3");
62
        saveas(gcf, "../Zad3/k2k3.png");
63
64
65 end
66
   function plot3_coeficients(src_vec)
67
        [k1,k2,k3] = get_coeficient_vectors(src_vec);
68
        plot3(k1,k2,k3, "*");
69
        xlabel("K1");
70
        ylabel("K2");
71
        zlabel("K3");
72
73 end
74
75
function [k1, k2, k3] = get_coeficient_vectors(src_vec)
78
        i = 1;
79
80
       for source = src_vec
 81
82
            disp(source);
83
            I1 = imread(source);
84
            I1 = double(I1) ./ 255;
85
 86
            if(size(I1,3) ~= 1)
87
                 I1 = rgb2gray(I1);
88
            end
89
90
91
            [k1(i), k2(i), k3(i)] = get_coeficients(I1);
92
93
            i = i + 1;
        end
94
95
96 end
97
98
99
100
   function [k1, k2, k3] = get_coeficients(I1)
102
103
        k1 = wspolczynnik_kompaktowosci(I1);
        k2 = wspolczynnik_malinowskiej(I1);
104
        k3 = wspolczynnik_Mz(I1);
105
106
107 end
```

$wspolczynnik_kompaktowosci.m$

```
function K = wspolczynnik_kompaktowosci(I1)

I1 = I1 > 0.5;

all1 = regionprops(I1, "all");

S = all1.Area;
L = all1.Perimeter;

K = 4*pi*S/L^2;

end
```

$wspolczynnik_malinowskiej.m$

```
function M = wspolczynnik_malinowskiej(I1)

I1 = I1 > 0.5;

all1 = regionprops(I1, "all");

S = all1.Area;
L = all1.Perimeter;

M = L / (2*sqrt(pi*S)) - 1;

and
end
```

$wspolczynnik_Mz.m$

```
function Mz = wspolczynnik_Mz(I1)

I1 = I1 > 0.5;

all1 = regionprops(I1, "all");

S = all1.Area;
L = all1.Perimeter;

Mz = 2 * sqrt(pi*S) / L;

end
```

g circles.m

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

g squares.m

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

g xd.m

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

$save_all.m$

```
clear all
close all
clc
save_circles
save_squares
save_xd
```

${\bf save_circles.m}$

```
1 clear all
_2 close all
3 clc
8 destination = "../images_gen";
9 mkdir(destination);
10
11 for i = 1:100;
12
13
       translation = (rand(1,2) - .5) * 200
      rotation = rand(1) * 360
scale = rand(1,2) * .3 + .5
14
15
16
       I1 = g_circles(translation, rotation, scale);
17
18
       filename = destination + "/circ" + string(i) + ".png";
imwrite(I1, filename);
19
20
21
22 end
```

save squares.m

```
1 clear all
_2 close all
3 clc
destination = "../images_gen";
7 mkdir(destination);
10 for i = 1:100;
11
     translation = (rand(1,2) - .5) * 200
12
     rotation = rand(1) * 360
scale = rand(1,2) * .3 + .5
13
14
15
     I1 = g_squares(translation, rotation, scale);
16
17
      filename = destination + "/sqr" + string(i) + ".png";
18
      imwrite(I1, filename);
19
20
21 end
```

$save_xd.m$

```
1 clear all
_2 close all
3 clc
7 destination = "../images_gen";
8 mkdir(destination);
10 for i = 1:100;
11
      translation = (rand(1,2) - .5) * 100
12
     rotation = rand(1) * 360
scale = rand(1,2) * .5 + 2
13
14
15
      I1 = g_xd(translation, rotation, scale);
16
17
18
       filename = destination + "/xD" + string(i) + ".png"; imwrite(I1, filename);
19
20
21
22 end
```