Politechnika Śląska

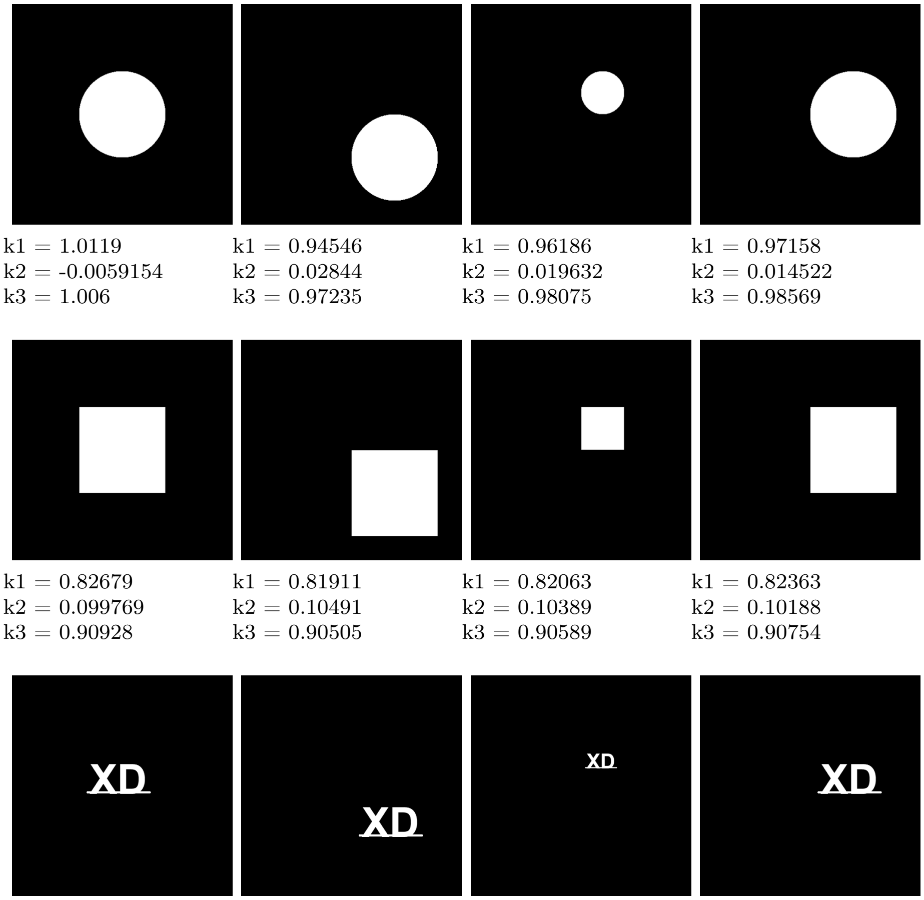
# Przetwarzanie Obrazów Cyfrowych

## Raport z ćwiczenia nr. 4: Klasyfikacja Obrazów

Raport opracował:
Dawid Kania
Grupa 6 Semestr 7

Data wykonania ćwiczenia: 12.12.2022

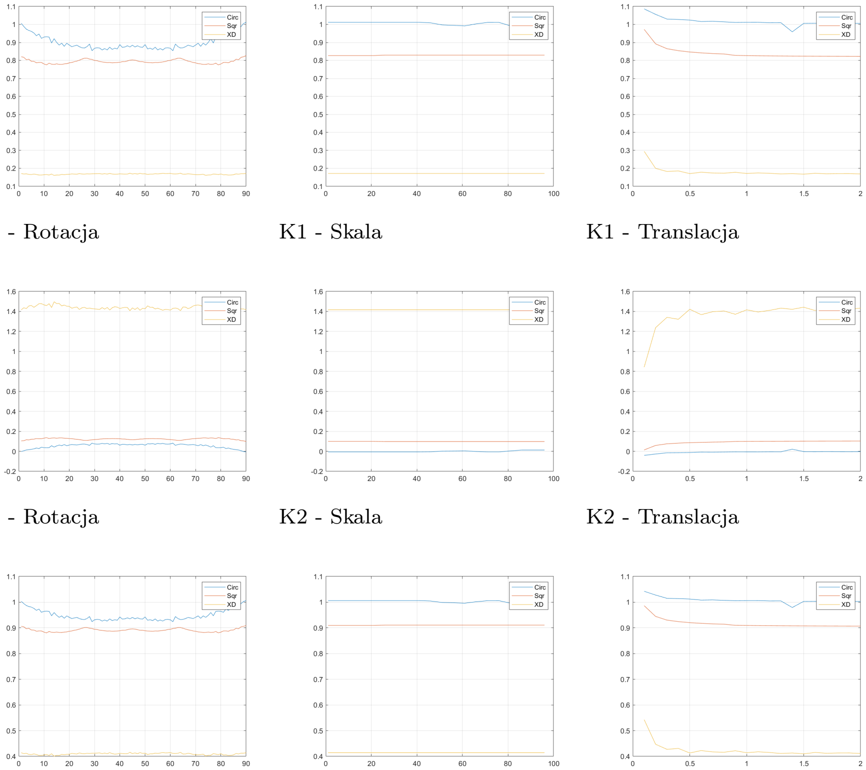
## Zadanie 2a



| | | | |
|---|---|---|---|
| k1 = 1.0119<br>k2 = -0.0059154<br>k3 = 1.006 | k1 = 0.94546<br>k2 = 0.02844<br>k3 = 0.97235 | k1 = 0.96186<br>k2 = 0.019632<br>k3 = 0.98075 | k1 = 0.97158<br>k2 = 0.014522<br>k3 = 0.98569 |
| k1 = 0.82679<br>k2 = 0.099769<br>k3 = 0.90928 | k1 = 0.81911<br>k2 = 0.10491<br>k3 = 0.90505 | k1 = 0.82063<br>k2 = 0.10389<br>k3 = 0.90589 | k1 = 0.82363<br>k2 = 0.10188<br>k3 = 0.90754 |
| k1 = 0.17126<br>k2 = 1.4164<br>k3 = 0.41384 | k1 = 0.17126<br>k2 = 1.4164<br>k3 = 0.41384 | k1 = 0.17048<br>k2 = 1.422<br>k3 = 0.41289 | k1 = 0.17126<br>k2 = 1.4164<br>k3 = 0.41384 |

Figure 1: Porownanie

# Zadanie 2b



K1 - Rotacja

K1 - Skala

K1 - Translacja

K2 - Rotacja

K2 - Skala

K2 - Translacja

K3 - Rotacja

K3 - Skala

K3 - Translacja

Figure 2: Porownanie

## Zadanie 3

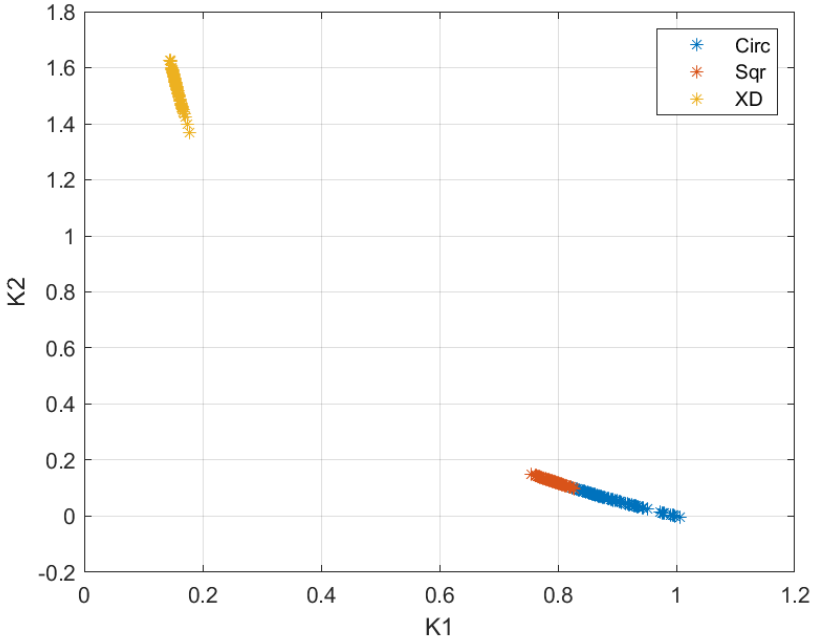


Figure 3: Porownanie

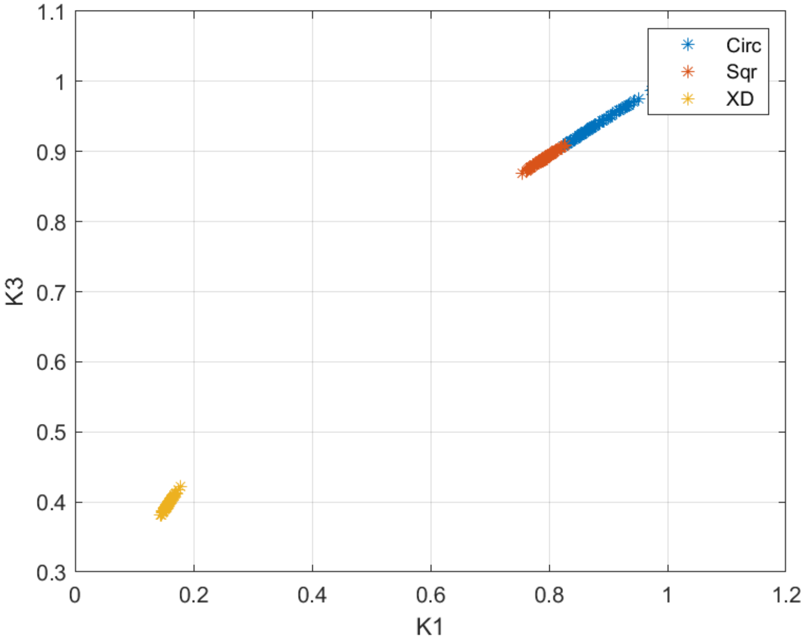


Figure 4: Porownanie

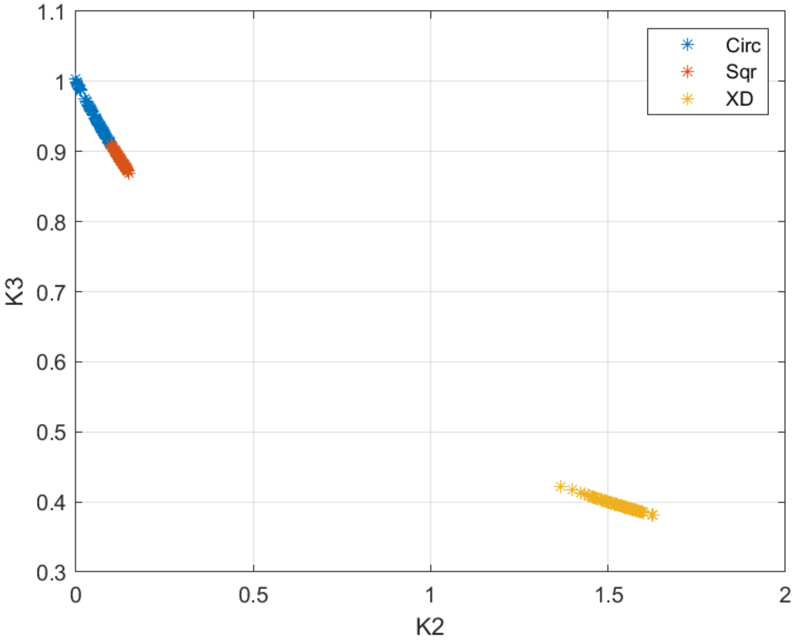


Figure 5: Porownanie

# Kody programów

## zad2a.m

```

clear all
close all
clc


figure

tiledlayout(3,4);


mkdir("../Zad2a/")
dest_folder = "../Zad2a";

K1 = zeros(12,1);
K2 = zeros(12,1);
K3 = zeros(12,1);

[K1( 1), K2( 1), K3( 1)] = show_img("../Zad2a/I1.png", @g_circles, [0 0],      0, [1 1]);
[K1( 2), K2( 2), K3( 2)] = show_img("../Zad2a/I2.png", @g_circles, [100 100], 0, [1 1]);
[K1( 3), K2( 3), K3( 3)] = show_img("../Zad2a/I3.png", @g_circles, [50 -50],  0, [.5 .5]);
[K1( 4), K2( 4), K3( 4)] = show_img("../Zad2a/I4.png", @g_circles, [100 0],   0, [1 1]);
[K1( 5), K2( 5), K3( 5)] = show_img("../Zad2a/I5.png", @g_squares, [0 0],      0, [1 1]);
[K1( 6), K2( 6), K3( 6)] = show_img("../Zad2a/I6.png", @g_squares, [100 100], 0, [1 1]);
[K1( 7), K2( 7), K3( 7)] = show_img("../Zad2a/I7.png", @g_squares, [50 -50],  0, [.5 .5]);
[K1( 8), K2( 8), K3( 8)] = show_img("../Zad2a/I8.png", @g_squares, [100 0],   0, [1 1]);
[K1( 9), K2( 9), K3( 9)] = show_img("../Zad2a/I9.png", @g_xd, [0 0],      0, [1 1]);
[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]);


Latex = [
    "\newcommand{\ww}{0.24} "
    "\begin{figure}[H] "
    "   \captionsetup[subfloat]{justification=raggedright, singlelinecheck=false, position=bottom,labelformat=empty} % "
    ""
    "    \subfloat[k1 = " + string(K1(1)) + " \\k2 = " + string(K2(1)) + "\\k3 = " + string(K3(1)) + " ]{"
    "        \includegraphics[width=\ww\linewidth]{" + dest_folder
    + "/I1.png}}  \hfill% "
    "    \subfloat[k1 = " + string(K1(2)) + " \\k2 = " + string(K2(2)) + "\\k3 = " + string(K3(2)) + " ]{"
    "        \includegraphics[width=\ww\linewidth]{" + dest_folder + "/I2.png}}  \hfill% "
    "    \subfloat[k1 = " + string(K1(3)) + " \\k2 = " + string(K2(3)) + "\\k3 = " + string(K3(3)) + " ]{"
    "        \includegraphics[width=\ww\linewidth]{" + dest_folder + "/I3.png}}  \hfill%"
    "    \subfloat[k1 = " + string(K1(4)) + " \\k2 = " + string(K2(4)) + "\\k3 = " + string(K3(4)) + " ]{"
    "        \includegraphics[width=\ww\linewidth]{" + dest_folder + "/I4.png}}  \hfill%"
    ""
    ""
    "    \subfloat[k1 = " + string(K1(5)) + " \\k2 = " + string(K2(5)) + "\\k3 = " + string(K3(5)) + " ]{"
    "        \includegraphics[width=\ww\linewidth]{" + dest_folder + "/I5.png}}  \hfill% "
    "    \subfloat[k1 = " + string(K1(6)) + " \\k2 = " + string(K2(6)) + "\\k3 = " + string(K3(6)) + " ]{"
    "        \includegraphics[width=\ww\linewidth]{" + dest_folder + "/I6.png}}  \hfill% "
    "    \subfloat[k1 = " + string(K1(7)) + " \\k2 = " + string(K2(7)) + "\\k3 = " + string(K3(7)) + " ]{"
    "        \includegraphics[width=\ww\linewidth]{" + dest_folder + "/I7.png}}  \hfill%"
    "    \subfloat[k1 = " + string(K1(8)) + " \\k2 = " + string(K2(8)) + "\\k3 = " + string(K3(8)) + " ]{"
    "        \includegraphics[width=\ww\linewidth]{" + dest_folder + "/I8.png}}  \hfill%"
    ""
    ""
    "    \subfloat[k1 = " + string(K1(9)) + " \\k2 = " + string(K2(9)) + "\\k3 = " + string(K3(9)) + " ]{"
    "        \includegraphics[width=\ww\linewidth]{" + dest_folder + "/I9.png}}  \hfill% "
    "    \subfloat[k1 = " + string(K1(10)) + " \\k2 = " + string(K2(10)) + "\\k3 = " + string(K3(10)) + " ]{"
    "        \includegraphics[width=\ww\linewidth]{" + dest_folder + "/I10.png}}  \hfill% "
    "    \subfloat[k1 = " + string(K1(11)) + " \\k2 = " + string(K2(11)) + "\\k3 = " + string(K3(11)) + " ]{"
    "        \includegraphics[width=\ww\linewidth]{" + dest_folder + "/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%"
    " "
    "\caption{Porownanie}"
    ""
    "\end{figure} "
    "\let\ww\undefined "
        ];

Latex = join(Latex,[''],2);
Latex = join(Latex,[newline],1);

fid = fopen(dest_folder + "/result.tex",’wt’);
fprintf(fid,"%s", Latex);
fclose(fid);






function [k1, k2, k3] = show_img(dest, func, trans, rot, scale)
    nexttile
    I1 = func(trans,rot,scale);
    imshow(I1);
    imwrite(I1, dest);
    k1 = wspolczynnik_kompaktowosci(I1);
    k2 = wspolczynnik_malinowskiej(I1);
    k3 = wspolczynnik_Mz(I1);
    title("K1 = " + string(k1) + " K2 = " + string(k2) + " K3 = " + string(k3));

end
```

### zad2b.m

```
clear all
close all
clc


mkdir("../Zad2b")

%%

[k1_circ, k2_circ, k3_circ, k1_sqr, k2_sqr, k3_sqr, k1_xd, k2_xd, k3_xd] = get_coeficients_rot()

plot_and_save("../Zad2b/k1_rot.png", [1:1:90] ,k1_circ, k1_sqr, k1_xd);
plot_and_save("../Zad2b/k2_rot.png", [1:1:90] ,k2_circ, k2_sqr, k2_xd);
plot_and_save("../Zad2b/k3_rot.png", [1:1:90] ,k3_circ, k3_sqr, k3_xd);

%%

[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);
plot_and_save("../Zad2b/k2_scale.png", [.1:.1:2] ,k2_circ, k2_sqr, k2_xd);
plot_and_save("../Zad2b/k3_scale.png", [.1:.1:2] ,k3_circ, k3_sqr, k3_xd);

%%

[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);
plot_and_save("../Zad2b/k2_trans.png", [1:5:100] ,k2_circ, k2_sqr, k2_xd);
plot_and_save("../Zad2b/k3_trans.png", [1:5:100] ,k3_circ, k3_sqr, k3_xd);


%%

dest_folder = "../Zad2b"

Latex = [
    "\newcommand{\ww}{0.32} "
    "\begin{figure}[H] "
    "   \captionsetup[subfloat]{justification=raggedright, singlelinecheck=false, position=bottom,labelformat=empty} % "
    ""
    "    \subfloat[K1 - Rotacja]{"
    "        \includegraphics[width=\ww\linewidth]{" + dest_folder + "/k1_rot.png}}  \hfill% "
    "    \subfloat[K1 - Skala]{"
    "        \includegraphics[width=\ww\linewidth]{" + dest_folder + "/k1_trans.png}}  \hfill% "
    "    \subfloat[K1 - Translacja]{"
    "        \includegraphics[width=\ww\linewidth]{" + dest_folder + "/k1_scale.png}}  \hfill% "
    ""
    "    \subfloat[K2 - Rotacja]{"
    "        \includegraphics[width=\ww\linewidth]{" + dest_folder + "/k2_rot.png}}  \hfill% "
    "    \subfloat[K2 - Skala]{"
    "        \includegraphics[width=\ww\linewidth]{" + dest_folder + "/k2_trans.png}}  \hfill% "
    "    \subfloat[K2 - Translacja]{"
    "        \includegraphics[width=\ww\linewidth]{" + dest_folder + "/k2_scale.png}}  \hfill% "
    ""
    "    \subfloat[K3 - Rotacja]{"
    "        \includegraphics[width=\ww\linewidth]{" + dest_folder + "/k3_rot.png}}  \hfill%"
    "    \subfloat[K3 - Skala]{"
    "        \includegraphics[width=\ww\linewidth]{" + dest_folder + "/k3_trans.png}}  \hfill%"
    "    \subfloat[K3 - Translacja]{"
    "        \includegraphics[width=\ww\linewidth]{" + dest_folder + "/k3_scale.png}}  \hfill%"
    ""
    ""
    ""
    "\caption{Porownanie}"
    ""
    "\end{figure} "
    "\let\ww\undefined "
        ];

Latex = join(Latex,[''],2);
Latex = join(Latex,[newline],1);

fid = fopen(dest_folder + "/result.tex",'wt');
fprintf(fid,"%s", Latex);
fclose(fid);




%%

function plot_and_save(dest,x,circ, sqr, xd)
    % K1 - rot
    hold off;
    plot(x, circ); hold on;
    plot(x, sqr);
    plot(x, xd);
    grid on;
    legend("Circ", "Sqr", "XD");
    saveas(gcf, dest);

end



function [k1_circ, k2_circ, k3_circ, k1_sqr, k2_sqr, k3_sqr, k1_xd, k2_xd, k3_xd] = get_coeficients_rot()

    translation = [0 0];
    scale = [1 1];
    all_rotations = [1:1:90];

    i = 1;
    for rotation = all_rotations;

        disp(rotation);

        I1 = g_circles(translation, rotation, scale );
        [k1_circ(i), k2_circ(i), k3_circ(i)] = get_coeficients(I1);

        I1 = g_squares(translation, rotation, scale );
        [k1_sqr(i), k2_sqr(i), k3_sqr(i)] = get_coeficients(I1);

        I1 = g_xd(translation, rotation, scale );
        [k1_xd(i), k2_xd(i), k3_xd(i)] = get_coeficients(I1);

        i = i + 1;
    end

end


function [k1_circ, k2_circ, k3_circ, k1_sqr, k2_sqr, k3_sqr, k1_xd, k2_xd, k3_xd] = get_coeficients_scale()

    translation = [0 0];
    all_scale = [.1:.1:2];
    rotation = 0;

    i = 1;
    for s1 = all_scale;

        scale = [s1, s1]

        disp(s1);

        I1 = g_circles(translation, rotation, scale );
        [k1_circ(i), k2_circ(i), k3_circ(i)] = get_coeficients(I1);

        I1 = g_squares(translation, rotation, scale );
        [k1_sqr(i), k2_sqr(i), k3_sqr(i)] = get_coeficients(I1);

        I1 = g_xd(translation, rotation, scale );
        [k1_xd(i), k2_xd(i), k3_xd(i)] = get_coeficients(I1);

        i = i + 1;
    end

end

function [k1_circ, k2_circ, k3_circ, k1_sqr, k2_sqr, k3_sqr, k1_xd, k2_xd, k3_xd] = get_coeficients_trans()

    all_translation = [1:5:100];
    scale = [1 1];
    rotation = 0;

    i = 1;
    for t1 = all_translation;

        translation = [t1 0];

        disp(t1);

        I1 = g_circles(translation, rotation, scale );
        [k1_circ(i), k2_circ(i), k3_circ(i)] = get_coeficients(I1);

        I1 = g_squares(translation, rotation, scale );
        [k1_sqr(i), k2_sqr(i), k3_sqr(i)] = get_coeficients(I1);

        I1 = g_xd(translation, rotation, scale );
        [k1_xd(i), k2_xd(i), k3_xd(i)] = get_coeficients(I1);

        i = i + 1;
    end

end



function [k1, k2, k3] = get_coeficients(I1)

    k1 = wspolczynnik_kompaktowosci(I1);
    k2 = wspolczynnik_malinowskiej(I1);
    k3 = wspolczynnik_Mz(I1);

end
```

### zad3.m

```

clear all
close all
clc



circ_source = "../images_gen/circ" + string(1:100) + ".png";
sqr_source = "../images_gen/sqr" + string(1:100) + ".png";
xD_source = "../images_gen/xD" + string(1:100) + ".png";




% plot3_coeficients(circ_source); hold on;
% plot3_coeficients(sqr_source);
% plot3_coeficients(xD_source);

mkdir("../Zad3")
plot_coeficients(circ_source, sqr_source, xD_source)



legend("Circ", "Sqr", "XD");
grid on;


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);
    [k1_v3,k2_v3,k3_v3] = get_coeficient_vectors(src_vec3);

    hold off;
    plot(k1_v1, k2_v1, '*'); hold on;
    plot(k1_v2, k2_v2, '*');
    plot(k1_v3, k2_v3, '*');
    grid on;
    legend("Circ", "Sqr", "XD");
    xlabel("K1");
    ylabel("K2");
    saveas(gcf, "../Zad3/k1k2.png");


    hold off;
    plot(k1_v1, k3_v1, '*'); hold on;
    plot(k1_v2, k3_v2, '*');
    plot(k1_v3, k3_v3, '*');
    grid on;
    legend("Circ", "Sqr", "XD");
    xlabel("K1");
    ylabel("K3");
    saveas(gcf, "../Zad3/k1k3.png");


    hold off;
    plot(k2_v1, k3_v1, ’*’); hold on;
    plot(k2_v2, k3_v2, ’*’);
    plot(k2_v3, k3_v3, ’*’);
    grid on;
    legend("Circ", "Sqr", "XD");
    xlabel("K2");
    ylabel("K3");
    saveas(gcf, "../Zad3/k2k3.png");

end

function plot3_coeficients(src_vec)
    [k1,k2,k3] = get_coeficient_vectors(src_vec);
    plot3(k1,k2,k3, "*");
    xlabel("K1");
    ylabel("K2");
    zlabel("K3");
end



function [k1, k2, k3] = get_coeficient_vectors(src_vec)

    i = 1;

    for source = src_vec

        disp(source);
        I1 = imread(source);
        I1 = double(I1) ./ 255;

        if(size(I1,3) ~= 1)
            I1 = rgb2gray(I1);
        end

        [k1(i), k2(i), k3(i)] = get_coeficients(I1);

        i = i + 1;
    end

end




function [k1, k2, k3] = get_coeficients(I1)

    k1 = wspolczynnik_kompaktowosci(I1);
    k2 = wspolczynnik_malinowskiej(I1);
    k3 = wspolczynnik_Mz(I1);

end
```

## wspolczynnik_kompaktowosci.m

```matlab
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

```matlab
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;

end
```

## wspolczynnik_Mz.m

```matlab
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

```matlab

function I1 = g_circles(translation, rotation, scale)

    % translation = [0 0];
    % rotation = [0];
    % scale = [0 0];


    canvas_size = [512, 512];
    radius = 100;
    center = canvas_size./2;

    % xy coordinates
    [x,y] = meshgrid(1:canvas_size(1), 1:canvas_size(2));
    x = x - center(1);
    y = y - center(2);

    % apply scale
    x = x ./ scale(1);
    y = y ./ scale(2);



    % draw circle
    I1 = (x.^2 + y.^2) < radius.^2;
    I1 = double(I1);

    I1 = imtranslate(I1, translation);
    I1 = imrotate(I1, rotation);


    %I1 = [ zeros(size(I1,1), 500), I1, zeros(size(I1,1), 500)];
    %I1 = [ zeros(500, size(I1,2)); I1; zeros(500, size(I1,2))];
    %cutted_part = ((scale-1).*canvas_size)/2 + 500;
    cutted_part = [(size(I1,1) - 512), (size(I1,2) - 512)]/2;
    crop_rect = [cutted_part,  511, 511];
    I1 = imcrop(I1, crop_rect);


    I1 = imresize(I1, [512, 512]);

end
```

### g_squares.m

```

function I1 = g_squares(translation , rotation , scale)

    % translation = [0 0];
    % rotation = [0];
    % scale = [0 0];


    canvas_size = [512, 512];
    size1 = 100;
    center = canvas_size./2;

    % xy coordinates
    [x,y] = meshgrid(1:canvas_size(1), 1:canvas_size(2));
    x = x - center(1);
    y = y - center(2);

    % apply scale
    x = x ./ scale(1);
    y = y ./ scale(2);



    % draw square
    I1 = (abs(x) < size1) & (abs(y) < size1);
    I1 = double(I1);

    I1 = imtranslate(I1, translation);
    I1 = imrotate(I1, rotation);


    cutted_part = [(size(I1,1) - 512), (size(I1,2) - 512)]/2;
    crop_rect = [cutted_part ,  511, 511];
    I1 = imcrop(I1, crop_rect);



    I1 = imresize(I1, [512, 512]);

end
```

## g_xd.m

```matlab

function I1 = g_xd(translation, rotation, scale)

    source = "../images/xD.png";

    % translation = [0 0];
    % rotation = [0];
    % scale = [0 0];

    I1 = imread(source);
    I1 = double(I1) ./ 255;

    if(size(I1,3) ~= 1)
        I1 = rgb2gray(I1);
    end

    canvas_size = [512, 512];
    size1 = 100;
    center = canvas_size./2;


    I1 = imrotate(I1, rotation);


    I1 = imresize(I1, scale.*canvas_size);



    I1 = [ zeros(size(I1,1), 500), I1, zeros(size(I1,1), 500)];
    I1 = [ zeros(500, size(I1,2)); I1; zeros(500, size(I1,2))];

    cutted_part = ((scale-1).*canvas_size)/2 + 500;

    crop_rect = [cutted_part,  511, 511];
    I1 = imcrop(I1, crop_rect);


    I1 = imtranslate(I1, translation);

    I1 = imresize(I1, [512, 512]);

end
```

## save_all.m

```
clear all
close all
clc


save_circles
save_squares
save_xd
```

### save_circles.m

```
clear all
close all
clc




destination = "../images_gen";
mkdir(destination);

for i = 1:100;

    translation = (rand(1,2) - .5) * 200
    rotation = rand(1) * 360
    scale = rand(1,2) * .3 + .5

    I1 = g_circles(translation, rotation, scale);

    filename = destination + "/circ" + string(i) + ".png";
    imwrite(I1, filename);

end
```

### save_squares.m

```
clear all
close all
clc


destination = "../images_gen";
mkdir(destination);


for i = 1:100;

    translation = (rand(1,2) - .5) * 200
    rotation = rand(1) * 360
    scale = rand(1,2) * .3 + .5

    I1 = g_squares(translation, rotation, scale);

    filename = destination + "/sqr" + string(i) + ".png";
    imwrite(I1, filename);

end
```

## save_xd.m

```
clear all
close all
clc



destination = "../images_gen";
mkdir(destination);

for i = 1:100;

    translation = (rand(1,2) - .5) * 100
    rotation = rand(1) * 360
    scale = rand(1,2) * .5 + 2

    I1 = g_xd(translation , rotation , scale);


    filename = destination + "/xD" + string(i) + ".png";
    imwrite(I1, filename);

end
```