

Gliwice, 08.11.2022

Przetwarzanie Obrazów Cyfrowych

Metody segmentacji obrazów



**Politechnika
Śląska**

Jakub Zeifert

1 Przygotowanie obrazów testowych

(Zaliczone w trakcie zajęć)

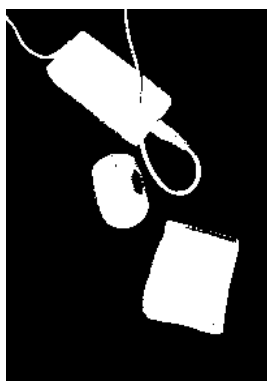


Rysunek 1: Własne obrazy użyte do testów

Są to zdjęcia przygotowane przeze mnie. Na pierwszym widoczna jest myszka od komputera, mysz oraz portfel. Na drugim zdjęciu ułożyłem kilka druków 3D wykonane z czarnego filamentu typu PLA.

2 Progowanie ze stałym (globalnym) progiem

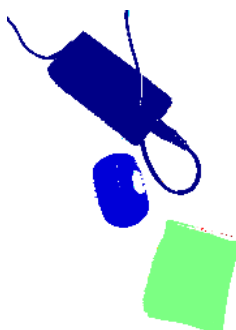
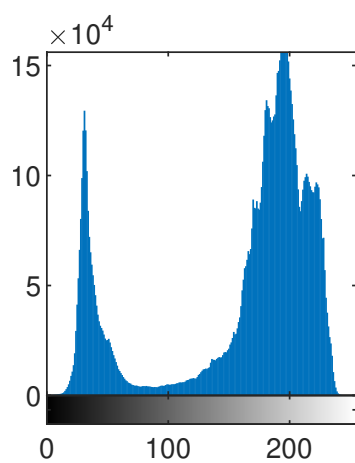
(Zaliczone w trakcie zajęć)



oryginał

Własny próg: 100

OTSU próg: 117



Ilość obszarów: 333

Ilość obszarów: 512

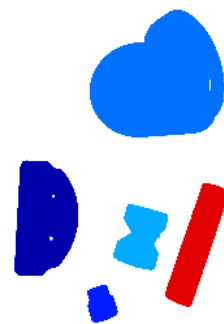
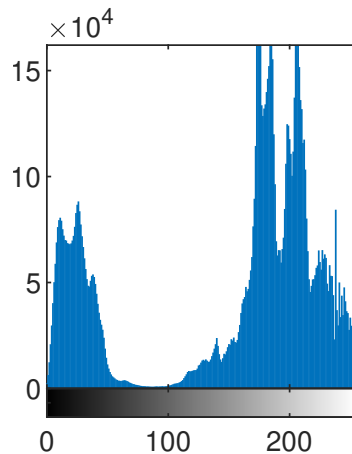
Rysunek 2: Progowanie ze stałym progiem



oryginał

Własny próg: 85

OTSU próg: 110



Ilość obszarów: 33

Ilość obszarów: 340

Rysunek 3: Progowanie ze stałym progiem

Użycie własnego progu dało mniejszą ilość obszarów, lecz na przy portfolio ze zdjęcia widać sporo zmian na krawędzi, gdzie odbijało się światło. W późniejszej części przy morfologii uda nam się pozbyć takiego problemu.

Wykorzystana funkcja do binaryzacji:

```

1 function result = binarize(img, threshold)
2 [X,Y,DIM] = size(img);
3 result = zeros(X,Y);
4 if threshold <= 0 || threshold >= 255
5     threshold = graythresh(img)*255; % OTSU
6 end
7 for i=1:1:X
8     for j=1:1:Y
9         if img(i,j) <= threshold
10             result(i,j) = 1;
11         else
12             result(i,j) = 0;
13         end
14     end
15 end
16 end
17 end

```

binarize.m

Wykorzystany kod:

```

1
2 clc
3 clear
4 %%

```

```

5 [file ,path]=uigetfile('*.','Select an image');
6 img = imread([path, file]);
7 img = rgb2gray(img);
8 % img_2 = imread("zdj\moje.jpg");
9 % figure(1);
10 % subplot(1,2,1);
11 % imshow(img);
12 % subplot(1,2,2);
13 % imshow(img_2);
14 % exportgraphics(figure(1),"orygina Ć.eps");
15 % img = rgb2gray(img);
16 plotter(img,85);
17
18 %% Functions
19
20 function plotter(img,treshold)
21     figure(1);
22     subplot(2,3,1);
23     imshow(img);
24     xlabel("orygina Ć");
25
26     subplot(2,3,2);
27     img_1 = bwlabel(binarize(img,treshold));
28 %     img_1 = ~img_1;
29     imshow(img_1);
30     xlabel(append("W Ćasny pr g: ",sprintf("%d",treshold)));
31
32     subplot(2,3,3);
33     img_2 = bwlabel(binarize(img,-1)); % -1 used to trigger OTSU
34 %     img_2 = ~img_2;
35     imshow(img_2);
36     xlabel(append("OTSU pr g: ",sprintf("%d",graythresh(img)*255)));
37     subplot(2,3,4);
38     imhist(img);
39
40     subplot(2,3,5);
41     imshow(label2rgb(img_1));
42     label_info = append("Ilo Ć obszar w: ",sprintf("%d",max(bwlabel(img_1),[],'all ')));
43     xlabel(label_info);
44
45     subplot(2,3,6);
46     imshow(label2rgb(img_2));
47     label_info = append("Ilo Ć obszar w: ",sprintf("%d",max(bwlabel(img_2),[],'all ')));
48     xlabel(label_info);
49     exportgraphics(figure(1),"zad1_2.eps");
50 end

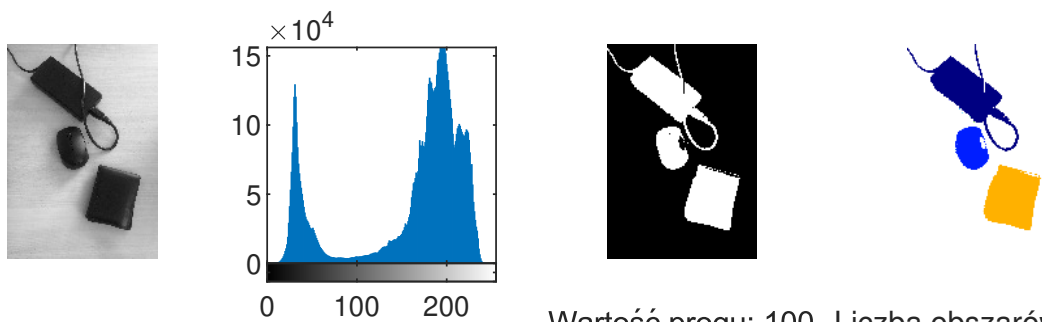
```

zad1.m

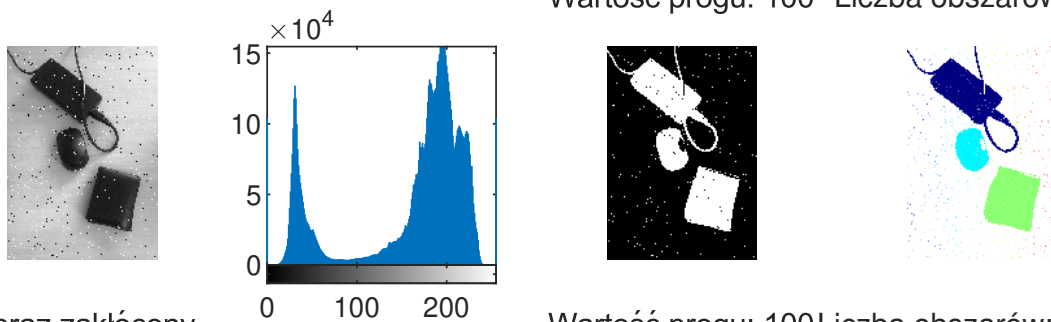
3 Wizualna ocena wyników binaryzacji

Wyniki metody OTSU stosunkowo dobrze binaryzuje obraz, lecz dostosowanie ręczne daje możliwość większej kontroli, a więc dokładniejszego dobrania progu. Własny próg umożliwia jeszcze lepsze odseparowanie konkretnych elementów i wyeliminowanie niepotrzebnych informacji z obrazów. Odpowiednie dobranie progu również umożliwia pozbycie się poszarpania konturów, które mogą występować między obiektem, który chcemy wyodrębnić a tłem. W zdjęciu numer dwa, dzięki zmniejszeniu progu udało się uzyskać lepszy podział elementów.

4 Wpływ przetwarzania wstępnego na binaryzację

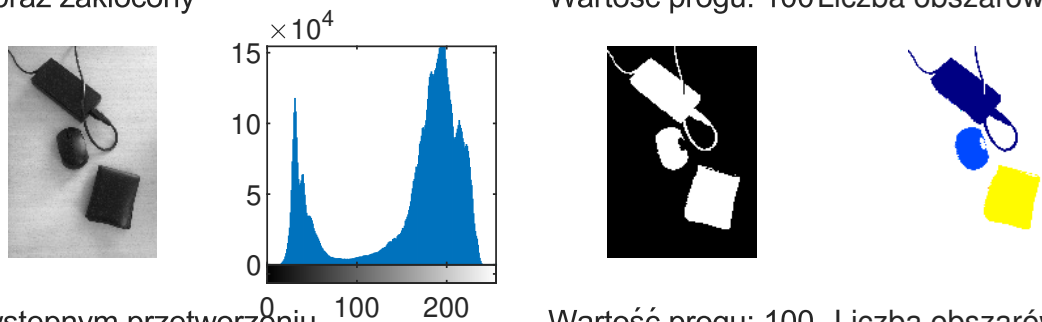


Wartość progu: 100 Liczba obszarów: 333



Obraz zakłócony

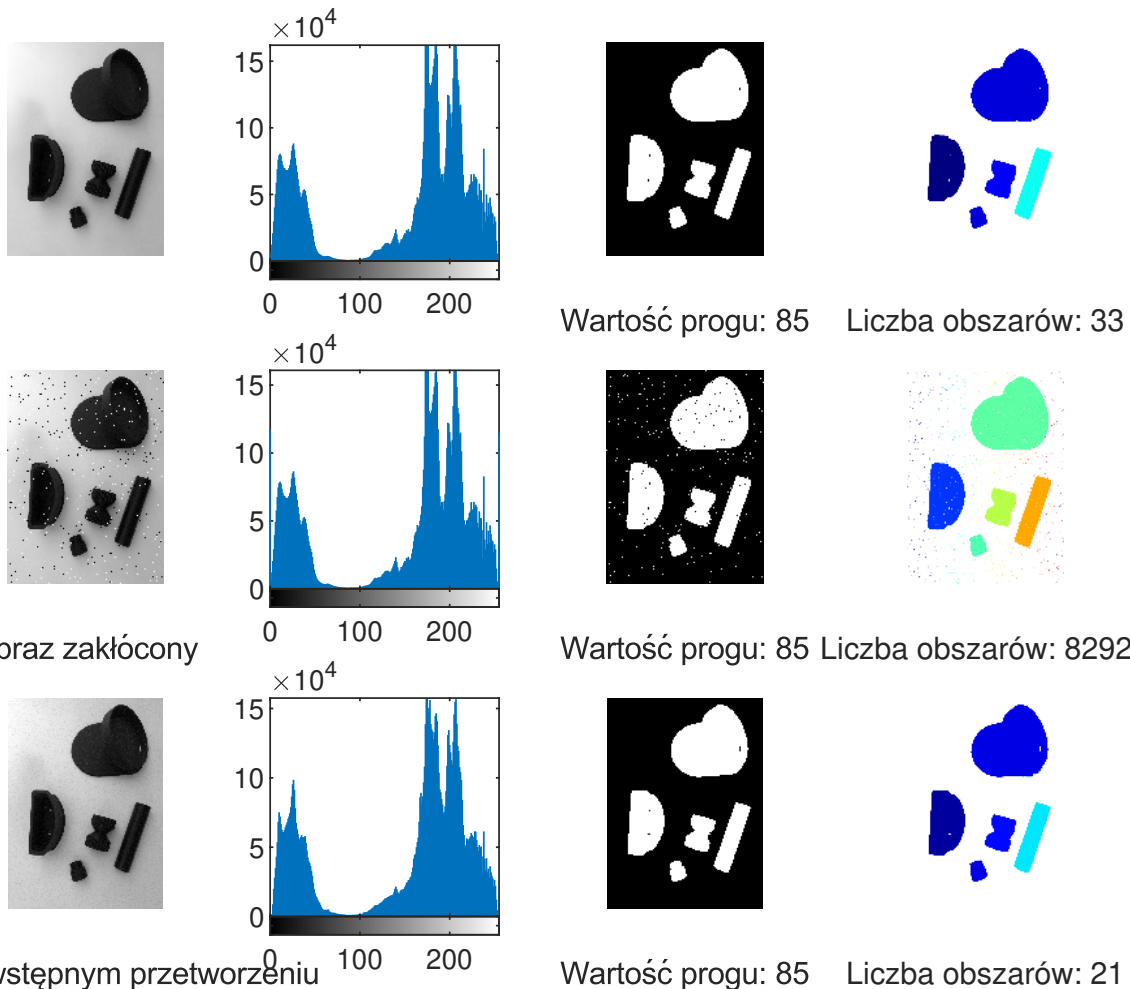
Wartość progu: 100Liczba obszarów: 80292



Obraz po wstępnym przetworzeniu

Wartość progu: 100 Liczba obszarów: 339

Rysunek 4: Wpływ przetwarzania wstępnego na binaryzację



Rysunek 5: Wpływ przetwarzania wstępnego na binaryzację

W przypadku obrazu pierwszego można zaobserwować, że zakłócenie obrazu zdecydowanie pogorszyło przejrzystość. Zastosowanie filtracji dolnoprzepustowej pozwoliło usunąć szum. Niestety liczba obszarów wzrosła nieznacznie, prawdopodobnie ze względu na zmiany niewielkiej ilości pikseli w tle. W przypadku obrazu drugiego liczba obszarów po usunięciu szumu zmniejszyła się. Jednakże wizualnie obraz binarny jest niemal identyczny do tego przed zakłóceniem.

Wykorzystany kod:

```

1
2 clc
3 clear
4 %%
5 [file ,path]=uigetfile({'*.*'}, 'Select an image');
6 img = imread([path, file]);
7
8
9 plotting(rgb2gray(img));
10
11
12
13
14
15 %% Functions
16
17
18 function plotting(img)
19     figure(1);
20     k = 1;
21     label_info = "";
22     zones = 1; % DOPISAE!!!! o co chodzi z t    liczba    obszar w
23     windowSize = 5;

```

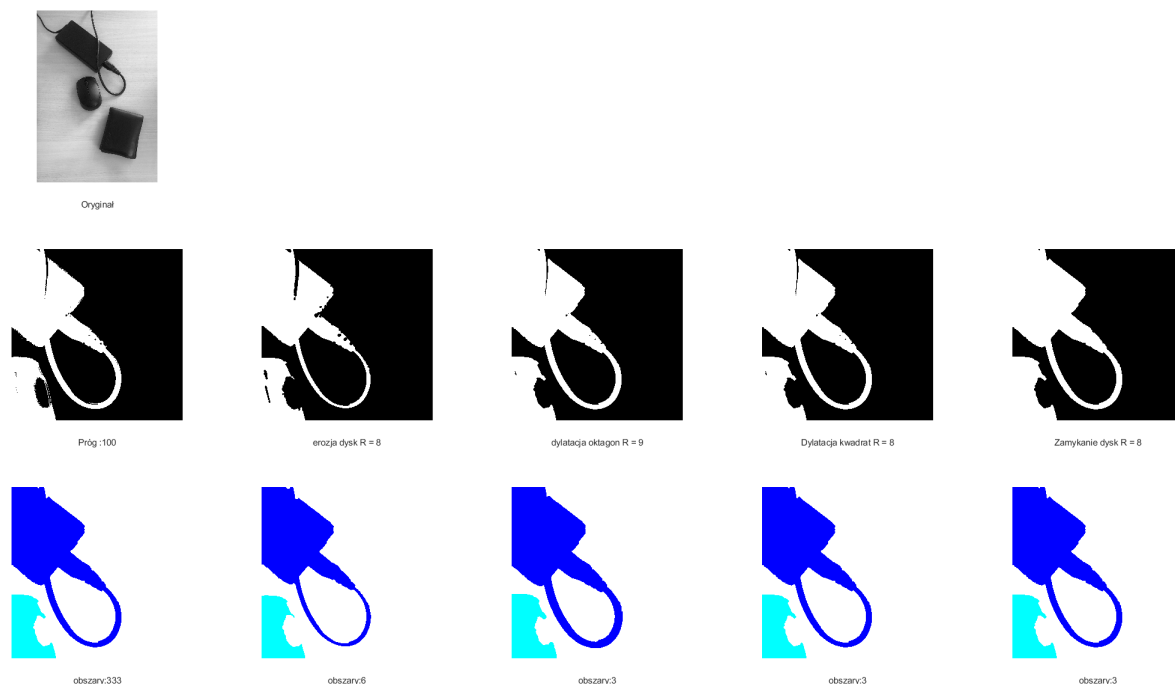
```

24     avg3 = ones(windowSize) / windowSize^2;
25 for i=1:1:3
26     switch i
27         case 2
28             img = imnoise(img,"salt & pepper",0.02);
29             label_info = "Obraz zak   cony";
30         case 3
31             img = imfilter(img,avg3,'conv');
32             label_info = "Obraz po wst  pnym przetworzeniu";
33     end
34     for j=1:1:4
35         subplot(3,4,k);
36         k = 1 + k;
37         switch j
38             case 1
39                 imshow(img);
40             case 2
41                 imhist(img);
42                 label_info = "";
43             case 3
44                 druk = 85;
45                 druk = 100;
46                 image_data = binarize(img,druk); % 100  adworka 85 druki
47                 label_info = append("Warto   progu: ",sprintf('%d',druk));
48                 imshow(image_data);
49             case 4
50                 image_data = bwlabel(binarize(img,-1));
51                 label_info =append("Liczba obszar w: ",sprintf('%d',max(
52                     bwlabel(binarize(img,druk)) ,[], 'all')));
53                 imshow(label2rgb(image_data));
54         end
55         xlabel(label_info);
56         exportgraphics(figure(1),"zad2_2.eps");
57     end
58 end

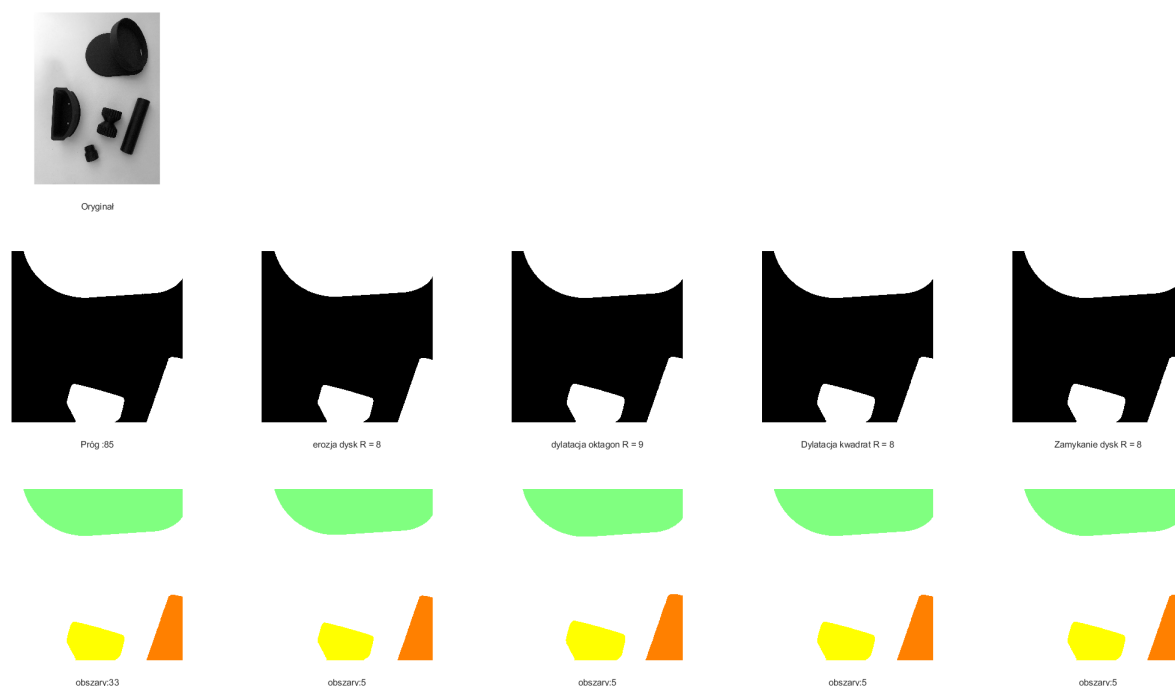
```

zad3.m

5 Zastosowanie operacji morfologicznych do poprawy wyników binaryzacji



Rysunek 6: Zastosowanie operacji morfologicznych do poprawy wyników binaryzacji



Rysunek 7: Zastosowanie operacji morfologicznych do poprawy wyników binaryzacji

Zastosowanie operacji morfologicznych w odpowiedniej kolejności poprawiło zdecydowanie liczbę obszarów widocznych na zdjęciu. Oznacza to usunięcie niepotrzebnych szumów oraz innych zakłóceń. Funkcja

Imerode pozwala na przeprowadzenie erozji zdjęcia. Jej działanie polega na obcinaniu brzegów obiektu na obrazie. Dylatacja służy do zamykania małych otworów oraz zatok we wnętrzu figury. Obiekty zwiększają swoją objętość i jeśli dwa lub więcej obiektów położonych jest blisko siebie, zrastają się w większe obiekty. Dzięki temu możemy pozbyć się rozwarstwień obiektu, który powinien być całością. Na koniec została przeprowadzone zamknięcie, a więc dylatacja a następnie erozja. Jako, że wcześniej wykonaliśmy erozję, a następnie dylatację (co w połączeniu nazywa się otwarciem), to nie został zaobserwowany żaden dodatkowy efekt.

Wykorzystany kod:

```
1 clc
2 clear
3
4 %%
5 [file ,path]=uigetfile('*.jpg','Select an image');
6 img = imread([path, file]);
7 img = rgb2gray(img);
8
9 plotting(img);
10
11 %% functions
12
13 function plotting(img)
14     figure(1);
15     k = 1;
16     label_info = "Oryginał ";
17     subplot(3,5,1);
18     imshow(img);
19     xlabel(label_info);
20 %     size_var_x = 600:2000;
21 %     size_var_y = 1100:2500;
22
23     size_var_x = 1100:2500;
24     size_var_y = 1100:2500;
25     for i=1:1:2
26         for j=1:1:5
27             switch j
28                 case 1
29                     if i==1
30                         result_img = binarize(img,85);
31                         bin_img = result_img;
32                         subplot(3,5,j+5);
33                         imshow(result_img(size_var_x,size_var_y));
34                         label_info = append("Próg :",sprintf("%d",85));
35                     else
36                         result_img_1 = label2rgb(bwlabel(result_img));
37                         subplot(3,5,j+10);
38                         imshow(result_img_1(size_var_x,size_var_y,:));
39                         label_info = append("obszary:",sprintf("%d",max(bwlabel(
40                             bin_img),[],'all')));
41                     end
42                 case 2
43                     if i==1
44                         r = 8;
45                         se = strel("disk",r);
46                         result_img = imerode(result_img,se);
47                         erode_img = result_img;
48                         subplot(3,5,j+5);
49                         imshow(result_img(size_var_x,size_var_y));
50                         label_info = append("erozja dysk R = ",sprintf("%d",r));
51                     else
52                         result_img_1 = label2rgb(bwlabel(imerode(result_img,se)));
53                         subplot(3,5,j+10);
54                         imshow(result_img_1(size_var_x,size_var_y,:));imshow(
```

```

54         result_img_1(size_var_x,size_var_y,:))
55         label_info = append("obszary:",sprintf("%d",max(bwlabel(
56             erode_img),[],'all')));
57     end
58     case 3
59         if i==1
60             r = 9;
61             se = strel("octagon",r);
62             result_img = imdilate(result_img,se);
63             dilate_img = result_img;
64             subplot(3,5,j+5);
65             imshow(result_img(size_var_x,size_var_y));
66             label_info = append("dylatacja oktagon R = ",sprintf("%d",r
67                 ));
68         else
69             result_img_1 = label2rgb(bwlabel(imdilate(result_img,se)));
70             subplot(3,5,j+10);
71             imshow(result_img_1(size_var_x,size_var_y,:));
72             label_info = append("obszary:",sprintf("%d",max(bwlabel(
73                 dilate_img),[],'all')));
74         end
75     case 4
76         if i==1
77             r = 8;
78             se = strel('line',r,r);
79             result_img = imopen(result_img,se);
80             open_img = result_img;
81             subplot(3,5,j+5);
82             imshow(result_img(size_var_x,size_var_y));
83             label_info = append("Dylatacja kwadrat R = ",sprintf("%d",r
84                 ));
85         else
86             result_img_1 = label2rgb(bwlabel(imopen(result_img,se)));
87             subplot(3,5,j+10);
88             imshow(result_img_1(size_var_x,size_var_y,:));
89             label_info = append("obszary:",sprintf("%d",max(bwlabel(
90                 open_img),[],'all')));
91         end
92     case 5
93         if i==1
94             r = 8;
95             se = strel('disk',r);
96             result_img = imclose(result_img,se);
97             close_img = result_img;
98             subplot(3,5,j+5);
99             imshow(result_img(size_var_x,size_var_y));
100            label_info = append("Zamykanie dysk R = ",sprintf("%d",r));
101        else
102            result_img_1 = label2rgb(bwlabel(imclose(result_img,se)));
103            subplot(3,5,j+10);
104            imshow(result_img_1(size_var_x,size_var_y,:));
105            label_info = append("obszary:",sprintf("%d",max(bwlabel(
106                close_img),[],'all')));
107        end
108    end
109    xlabel(label_info);
110    exportgraphics(fgure(1),"zad3_2.eps")
111 end
112 end
113 end

```