

**Министерство науки и высшего образования
Российской Федерации**

**Федеральное государственное автономное
образовательное учреждение высшего образования**

**«Национальный исследовательский университет
ИТМО»**

**Факультет информационных технологий и
программирования**

Лабораторная работа № 6

Сегментация объектов на изображении

Выполнил студент группы № М3302

Суворин Ярослав Владимирович

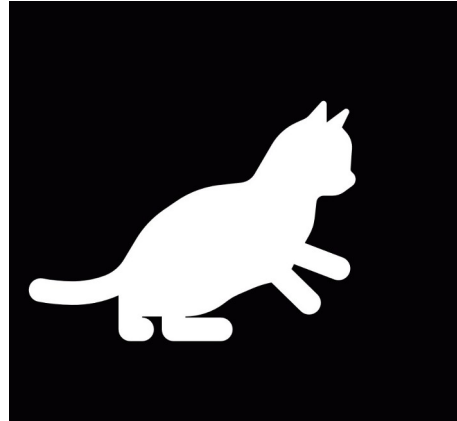
Санкт-Петербург
2025

1. Цель работы

Приобретение навыков сегментации объектов на цифровом изображении с использованием разных подходов.

2. Ход выполнения работы

2.a Исходные изображения



2.b Листинги программных реализаций

```
img = imread('rat1.jpg');

x = [350, 500, 700, 500, 350];
y = [350, 200, 400, 700, 400];

[rows, cols, ~] = size(img);
[X, Y] = meshgrid(1:cols, 1:rows);
mask = inpolygon(X, Y, x, y);

masked_img = bsxfun(@times, img, uint8(mask));

subplot(1,2,1); imshow(img); title('Исходное изображение');
subplot(1,2,2); imshow(masked_img); title('Вырезанный пятиугольник');
```

```
originalImg = imread('rat1.jpg');
maskImg = imread('ЛР6_1.jpg');

maskImg = rgb2gray(maskImg);
binaryMask = imbinarize(maskImg);
binaryMask = imresize(binaryMask, [size(originalImg, 1), size(originalImg, 2)]);
segmentedImg = bsxfun(@times, originalImg, uint8(binaryMask));

subplot(1,3,1); imshow(originalImg); title('Исходное изображение');
subplot(1,3,2); imshow(maskImg); title('Маска');
subplot(1,3,3); imshow(segmentedImg); title('Сегментированное изображение');
```

```

img = imread('rat1.jpg');
img = rgb2gray(img);

global_thresh = graythresh(img);
binary_global = imbinarize(img, global_thresh);

subplot(1,3,1); imshow(img); title('Исходное изображение');
subplot(1,3,2); imshow(binary_global); title('Глобальная бинаризация (Оцу)');

block_size = 32;
binary_local = zeros(size(img));
for i = 1:block_size:size(img, 1)
    for j = 1:block_size:size(img, 2)
        row_end = min(i + block_size - 1, size(img, 1));
        col_end = min(j + block_size - 1, size(img, 2));
        block = img(i:row_end, j:col_end);
        local_thresh = graythresh(block);
        binary_local(i:row_end, j:col_end) = imbinarize(block, local_thresh);
    end
end

subplot(1,3,3); imshow(binary_local); title('Локальная бинаризация (Оцу по блокам)');

```

```

img = imread('rat1.jpg');
img = rgb2gray(img);

sobel_x = [-1 0 1; -2 0 2; -1 0 1];
sobel_y = [-1 -2 -1; 0 0 0; 1 2 1];
Gx = imfilter(double(img), sobel_x, 'same');
Gy = imfilter(double(img), sobel_y, 'same');
gradient_magnitude = sqrt(Gx.^2 + Gy.^2);
gradient_normalized = mat2gray(gradient_magnitude);
binary_edges = imbinarize(gradient_normalized, 0.1);

subplot(1,3,1); imshow(img); title('Исходное');
subplot(1,3,2); imshow(binary_edges); title('Границы (Собель)');

edges_canny = edge(img, 'canny', [0.1 0.2]);
subplot(1,3,3); imshow(edges_canny); title('Границы (Кенни)');

```

```

img = imread('rat1.jpg');
img = rgb2gray(img);

seeds = [100, 100; 200, 200; 300, 300; 400, 400; 500, 500];
delta_thresh = 30;
[segmented, boundaries] = region_growing(img, seeds, delta_thresh);

subplot(1,2,1); imshow(img); title('Исходное изображение');
subplot(1,2,2); imshow(label2rgb(segmented)); title('Результат сегментации');

function [labeled_image, boundaries] = region_growing(image, seeds, delta_threshold)
    [height, width] = size(image);
    labeled_image = zeros(height, width);
    boundaries = false(height, width);

    n_regions = size(seeds, 1);
    region_means = zeros(n_regions, 1);

    for i = 1:n_regions
        x = seeds(i, 1);
        y = seeds(i, 2);
        labeled_image(y, x) = i;
        region_means(i) = image(y, x);
    end

    listB = zeros(0, 4, 'uint32');
    visited = false(height, width);

    for i = 1:n_regions
        x = seeds(i, 1);
        y = seeds(i, 2);
        neighbors = get_neighbors(x, y, height, width);

        for j = 1:size(neighbors, 1)
            nx = neighbors(j, 1);
            ny = neighbors(j, 2);
            if labeled_image(ny, nx) == 0 && ~boundaries(ny, nx) && ~visited(ny, nx)
                delta = abs(image(ny, nx) - region_means(i));
                listB = [listB; uint32(nx), uint32(ny), delta, i];
                visited(ny, nx) = true;
            end
        end
    end

    listB = sortrows(listB, 3);

    while ~isempty(listB)
        current = listB(1, :);
        listB(1, :) = [];

        x = current(1);
        y = current(2);
        region_id = current(4);
        neighbors = get_neighbors(x, y, height, width);
        neighbor_regions = [];

        for j = 1:size(neighbors, 1)
            nx = neighbors(j, 1);
            ny = neighbors(j, 2);
            nr = labeled_image(ny, nx);
            if nr > 0
                neighbor_regions = [neighbor_regions; nr];
            end
        end

        neighbor_regions = unique(neighbor_regions);

        if numel(neighbor_regions) == 1 && neighbor_regions == region_id
            labeled_image(y, x) = region_id;
        end
    end
end

```

```

region_pixels = image(labeled_image == region_id);
region_means(region_id) = mean(region_pixels);

neighbors = get_neighbors(x, y, height, width);

for j = 1:size(neighbors, 1)
    nx = neighbors(j, 1);
    ny = neighbors(j, 2);

    if labeled_image(ny, nx) == 0 && ~boundaries(ny, nx) && ~visited(ny, nx)
        delta = abs(image(ny, nx) - region_means(region_id));
        if delta <= delta_threshold
            listB = [listB; nx, ny, delta, region_id];
            visited(ny, nx) = true;
        end
    end
end

listB = sortrows(listB, 3);
else
    boundaries(y, x) = true;
end
end
end

function neighbors = get_neighbors(x, y, height, width)
    neighbors = [];
    |
    if x > 1, neighbors = [neighbors; x-1, y]; end
    if x < width, neighbors = [neighbors; x+1, y]; end
    if y > 1, neighbors = [neighbors; x, y-1]; end
    if y < height, neighbors = [neighbors; x, y+1]; end
end

```

```

img = imread('rat1.jpg');
img = rgb2gray(img);
[labels, fg_markers, bg_markers] = watershed_segmentation(img);

imshow(label2rgb(labels, 'jet', 'k', 'shuffle'));
title('Окончательная сегментация');
]
function [labels, foreground_markers, background_markers] = watershed_segmentation(img)
    se = strel('disk', 5);
    img = imclose(img, se);

    bw = imbinarize(img);
    D = bwdist(~bw);

    mask = imextendedmax(D, 2);

    foreground_markers = bwlabel(mask);
    num_objects = max(foreground_markers(:));

    background_markers = ~bw;
    background_markers = imdilate(background_markers, strel('disk', 3));

    markers = foreground_markers;
    markers(background_markers) = num_objects + 1;

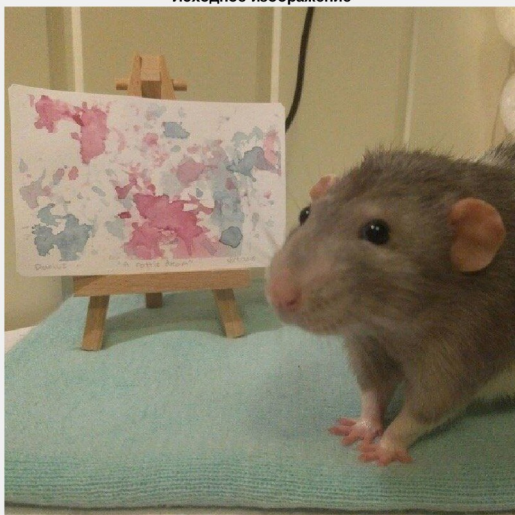
    D = imcomplement(D);
    D = imimposemin(D, markers);
    labels = watershed(D);

    subplot(2,2,1); imshow(img); title('Исходное изображение');
    subplot(2,2,2); imshow(D,[]); title('Преобразование расстояния');
    subplot(2,2,3); imshow(label2rgb(markers)); title('Маркеры');
    subplot(2,2,4); imshow(label2rgb(labels)); title('Результат сегментации');
end

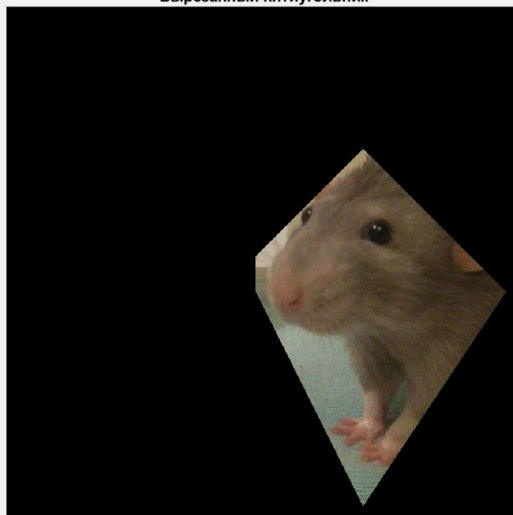
```

2.d Результирующие изображения

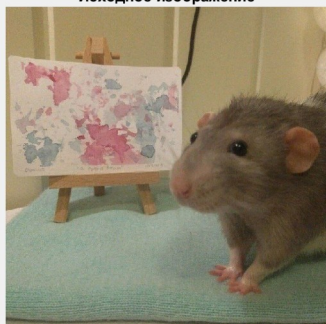
Исходное изображение



Вырезанный пятиугольник



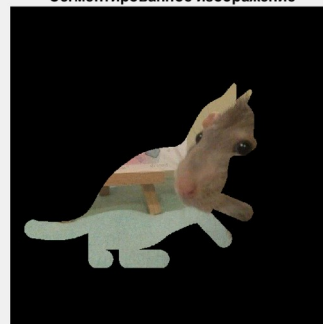
Исходное изображение



Маска



Сегментированное изображение



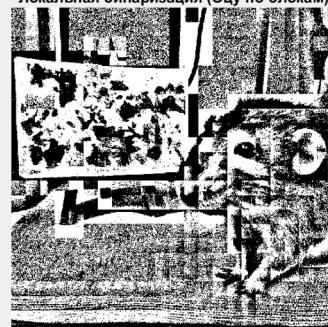
Исходное изображение



Глобальная бинаризация (Оцу)



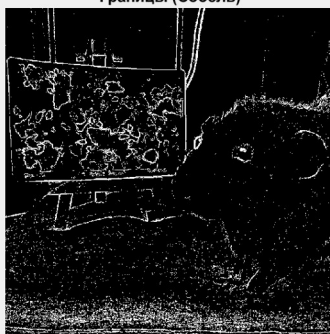
Локальная бинаризация (Оцу по блокам)



Исходное



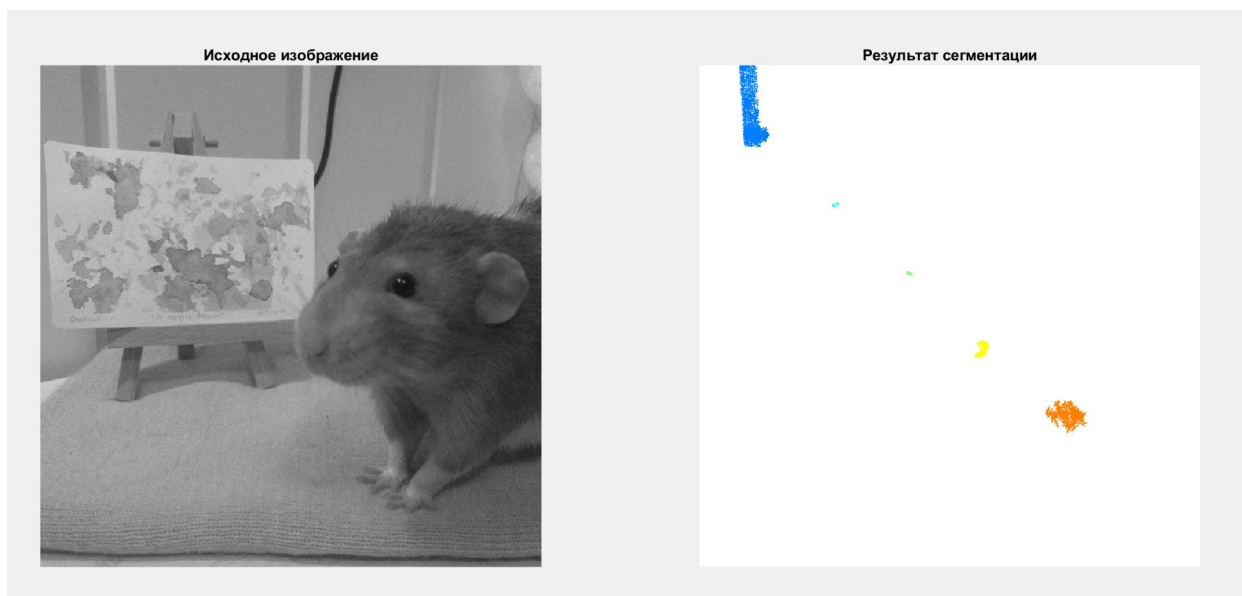
Границы (Собель)



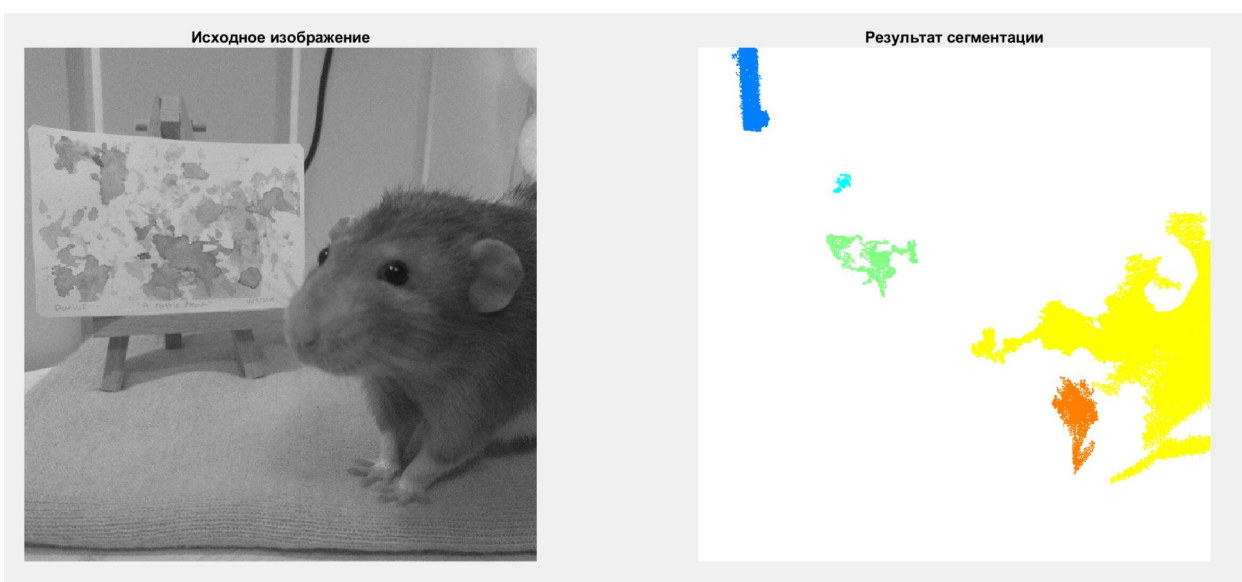
Границы (Кенни)



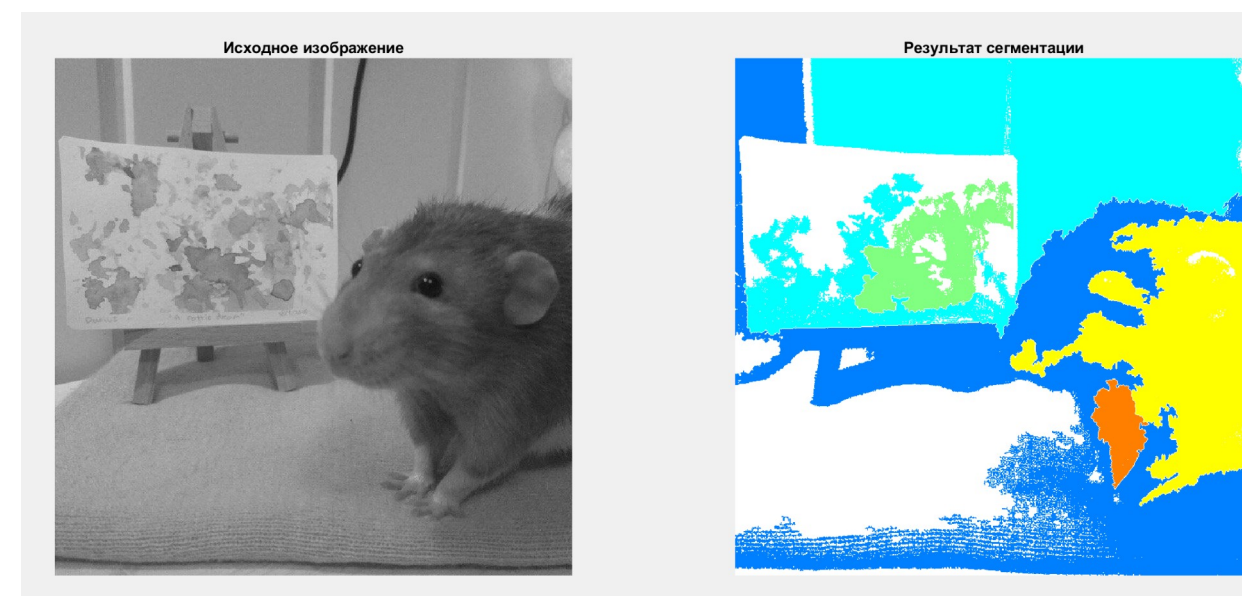
threshold = 1

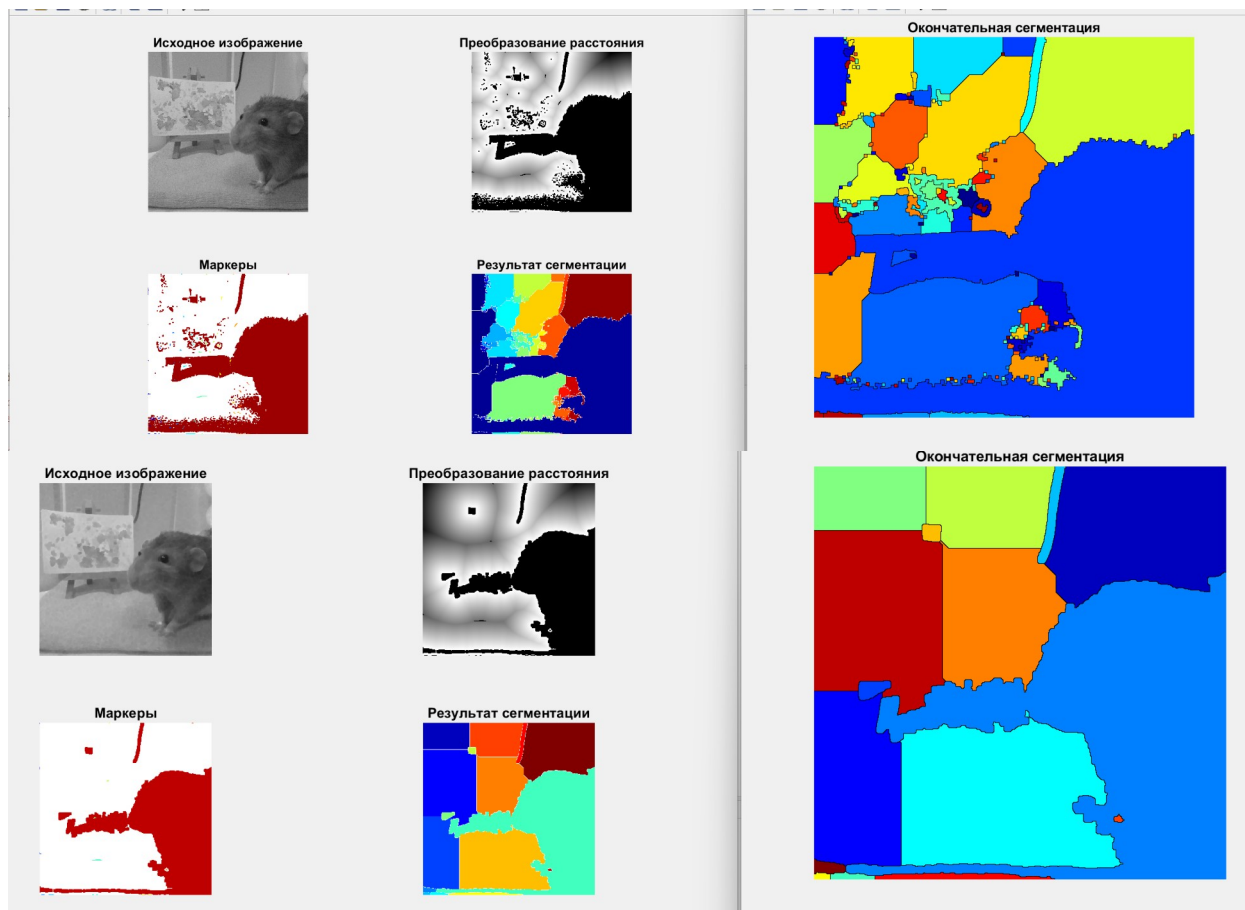


threshold = 10



threshold = 30





3. Вывод по работе

В результате проведенной работы я сегментировал изображение разными способами — начиная от самых простых (пятиугольник, вырезание по форме) и заканчивая алгоритмами, такими как выращивание областей и водораздел. Можно сравнить эффективность последних двух методов — видно, что в обоих случаях сегментации не смогли разделить сливающийся нос крысы и тень от холста, в результате чего они выделились в один сегмент. Пятнам на холсте видно, что выращивание смогло их определить (в разных пропорциях, в зависимости от threshold), а вот водораздел не смог их разделить. Также интересно будет заметить влияние морфологического закрытия в водоразделе — с одной стороны, оно очистило итоговый результат от лишних сегментов, с другой стороны — испортило форму сегмента подставки холста.