

**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 № 1

Преобразования к одномерным инструментам анализа

Выполнил студент группы № М3302

Суворин Ярослав Владимирович

Санкт-Петербург
2025

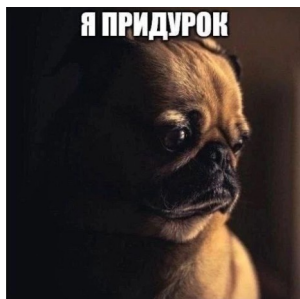
1. Цель работы

Получение основных навыков работы с цифровым изображением и освоение основных яркостных и геометрических характеристик изображений.

2. Ход выполнения работы

2.а Исходные изображения

Исходные изображения для Части 1:



Исходные изображения для Части 2:

Сильно и слабоконтрастные изображения для построения гистограммы:



Изображение с монотонными и выделяющимися объектами для построения проекции:

наши дни сочтены



Изображения — увеличенные строчки текста и штрих код для построения профиля:
вдоль изображения.



2.b Листинги программных реализаций

Код для выполнения Части 1:

```
dog=imread('dog.jpg');

subplot(3, 3, 1)
imshow(dog)
title('original image')

subplot(3, 3, 2)
dog_gray=rgb2gray(dog);
imshow(dog_gray)
title('grayscale')

subplot(3, 3, 3)
dog_red = uint8(zeros(size(dog)));
dog_red(:,:,1) = dog(:,:,1);
imshow(dog_red);
title('RED extracted')

subplot(3, 3, 4)
dog_green = uint8(zeros(size(dog)));
dog_green(:,:,2) = dog(:,:,2);
imshow(dog_green);
title('GREEN extracted')

subplot(3, 3, 5)
dog_blue = uint8(zeros(size(dog)));
dog_blue(:,:,3) = dog(:,:,3);
imshow(dog_blue);
title('BLUE extracted')

subplot(3, 3, 6)
R = double(dog(:, :, 1));
G = double(dog(:, :, 2));
B = double(dog(:, :, 3));
dog_gray_halfitone=(R+G+B)/3;
dog_gray_halfitone=uint8(dog_gray_halfitone);
imshow(dog_gray_halfitone);
title('halftone')

subplot(3, 3, 7)
dog_binary_25 = dog_gray_halfitone > 0.25*255;
imshow(dog_binary_25);
title('binary 25%')

subplot(3, 3, 8)
dog_binary_50 = dog_gray_halfitone > 0.5*255;
imshow(dog_binary_50);
title('binary 50%')

subplot(3, 3, 9)
dog_binary_75 = dog_gray_halfitone > 0.75*255;
imshow(dog_binary_75);
title('binary 75%')
```

Код для выполнения Части 2 пункта 1 — построения гистограмм:

```
image_low=imread("low_contrast.jpg");
subplot(2, 3, 1);
imhist(image_low(:,:,1));
title('low contrast red');
subplot(2, 3, 2);
imhist(image_low(:,:,2));
title('low contrast green');
subplot(2, 3, 3);
imhist(image_low(:,:,3));
title('low contrast blue');

image_high=imread("high_contrast.jpg");
subplot(2, 3, 4);
imhist(image_high(:,:,1));
title('high contrast red');
subplot(2, 3, 5);
imhist(image_high(:,:,2));
title('high contrast green');
subplot(2, 3, 6);
imhist(image_high(:,:,3));
title('high contrast blue');
```

Код для выполнения Части 2 пункта 2 — построения проекций и обозначения границ объектов с помощью пороговых значений:

```
image_stand_out=imread("stand_out.jpg");
[numRows, numCols, Layers] = size(image_stand_out);

subplot(3, 1, 1);
horizontal_projection = zeros(numCols, 1);
for i=1:numCols
    horizontal_projection(i,1)=(round(sum(image_stand_out(:,i,1)))+round(sum(image_stand_out(:,i,2)))+round(sum(image_stand_out(:,i,3))))/(256*3);
end
plot(horizontal_projection);
title('horizontal projection')

subplot(3, 1, 2);
vertical_projection = zeros(numRows, 1);
for i=1:numRows
    vertical_projection(i,1)=(round(sum(image_stand_out(i,:,1)))+round(sum(image_stand_out(i,:,2)))+round(sum(image_stand_out(i,:,3))))/(256*3);
end
plot(vertical_projection);
title('vertical projection')

subplot(3, 1, 3);
imshow(image_stand_out);
threshold = 0.999 * max(horizontal_projection);
horizontal_boundaries = [];
for i = 2:numCols
    if horizontal_projection(i) < threshold && horizontal_projection(i-1) >= threshold
        horizontal_boundaries = [horizontal_boundaries; i];
    elseif horizontal_projection(i) >= threshold && horizontal_projection(i-1) < threshold
        horizontal_boundaries = [horizontal_boundaries; i];
    end
end

for i = 1:length(horizontal_boundaries)
    line([horizontal_boundaries(i) horizontal_boundaries(i)], [1 numRows], 'Color', 'b');
end

threshold = 0.999 * max(vertical_projection);
vertical_boundaries = [];
for i = 2:numRows
    if vertical_projection(i) < threshold && vertical_projection(i-1) >= threshold
        vertical_boundaries = [vertical_boundaries; i];
    elseif vertical_projection(i) >= threshold && vertical_projection(i-1) < threshold
        vertical_boundaries = [vertical_boundaries; i];
    end
end

for i = 1:length(vertical_boundaries)
    line([1 numCols], [vertical_boundaries(i) vertical_boundaries(i)], 'Color', 'b');
end
```

Код для выполнения Части 2 пункта 3 — построения профиля:

```
text=imread('text.jpg');
subplot(2, 1, 1);
[numRows, numCols, Layers]=size(text);
x=[1 numCols];
y=[ceil(numRows/2) ceil(numRows/2)];
improfile(text,x,y),grid on;
title('text');

code=imread('code.jpg');
subplot(2, 1, 2);
[numRows_c, numCols_c, Layers_c]=size(text);
x_c=[1 numCols_c];
y_c=[ceil(numRows_c/2) ceil(numRows_c/2)];
improfile(code,x_c,y_c),grid on;
title('code');|
```

Код для выполнения дополнительного задания(финальный результат):

```
image_low=imread("low_contrast.jpg");

image_equalized_histeq = zeros(size(image_low), 'uint8');
for channel = 1:3
    image_equalized_histeq(:, :, channel) = histeq(image_low(:, :, channel));
end

image_low(image_low <= 95) = 95;
[numRows, numCols, Layers] = size(image_low);
for k=1:1:Layers
    image_min = min(min(image_low(:, :, k)));
    image_max = max(max(image_low(:, :, k)));
    for i=1:1:numRows
        for j=1:1:numCols
            element = (double(image_low(i,j,k) - image_min) / double(image_max - image_min));
            image_equalized_my(i,j,k) = element^0.5;
            if image_equalized_my(i,j,k) > 1
                image_equalized_my(i,j,k) = 1;
            end
            if image_equalized_my(i,j,k) < 0
                image_equalized_my(i,j,k) = 0;
            end
        end
    end
end

subplot(4, 3, 1);
imhist(image_low(:, :, 1));
title('Исходный красный канал');
subplot(4, 3, 2);
imhist(image_low(:, :, 2));
title('Исходный зеленый канал');
subplot(4, 3, 3);
imhist(image_low(:, :, 3));
title('Исходный синий канал');

subplot(4, 3, 4);
imhist(image_equalized_histeq(:, :, 1));
title('Выравненный красный канал (histeq)');
subplot(4, 3, 5);
imhist(image_equalized_histeq(:, :, 2));
title('Выравненный зеленый канал (histeq)');
subplot(4, 3, 6);
imhist(image_equalized_histeq(:, :, 3));
title('Выравненный синий канал (histeq)');

subplot(4, 3, 7);
imhist(image_equalized_my(:, :, 1));
title('Выравненный красный канал (вручную)');
subplot(4, 3, 8);
imhist(image_equalized_my(:, :, 2));
title('Выравненный зеленый канал (вручную)');
subplot(4, 3, 9);
imhist(image_equalized_my(:, :, 3));
title('Выравненный синий канал (вручную)');

subplot(4, 3, 10);|
imshow(image_low)
subplot(4, 3, 11);
imshow(image_equalized_histeq)
subplot(4, 3, 12);
imshow(image_equalized_my)
```

2.с Комментарии

Комментарии по Части 1:

В результирующем изображении выведено два серых изображения — одно с помощью `rgb2gray`, другое с помощью усреднения интенсивности по цветовым каналам:

```
R = double(dog(:, :, 1));  
G = double(dog(:, :, 2));  
B = double(dog(:, :, 3));
```

Разница в том, что, в отличие от усреднения, `rgb2gray` выставляет собственные коэффициенты для каждого цвета:

$0.299 * R + 0.587 * G + 0.114 * B$

Комментарии по Части 2:

Пункт 1:

Изобразил гистограммы для слабоконтрастного изображения. Для сравнения изобразил и для сильноконтрастного: на гистограммах видно, что диапазон серого у сильноконтрастного больше.

Пункт 2:

После написания кода и вывода графиков можно сопоставить графики и само изображение и увидеть четкие границы выделяющихся объектов:



Также с помощью пороговых значений обозначил выделение объектов на изображении. Видно, что как и „колобок“, так и текст, выделяются. Однако, по горизонтальному отображению с помощью пороговых значений не удастся обозначить начало текста — так как по оси X он пересекается с „колобком“.

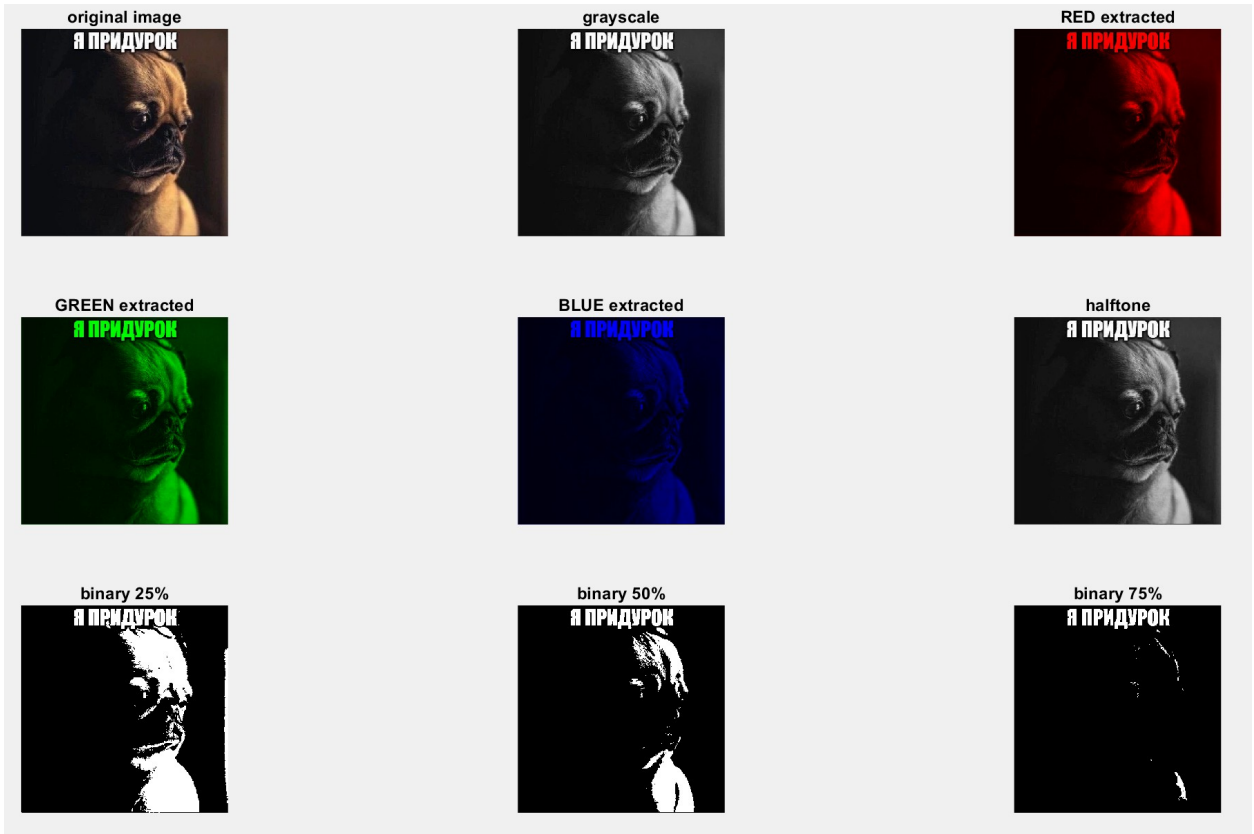
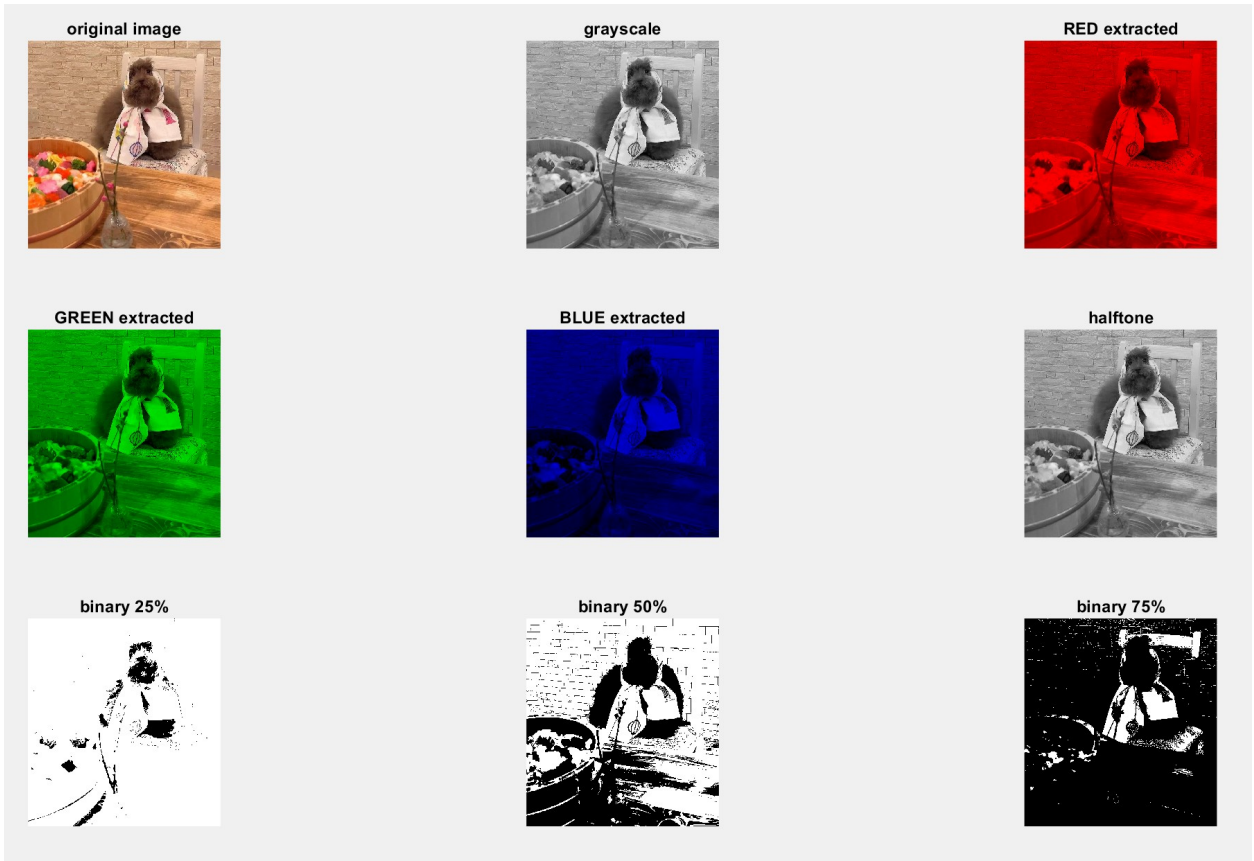


Пункт 3:

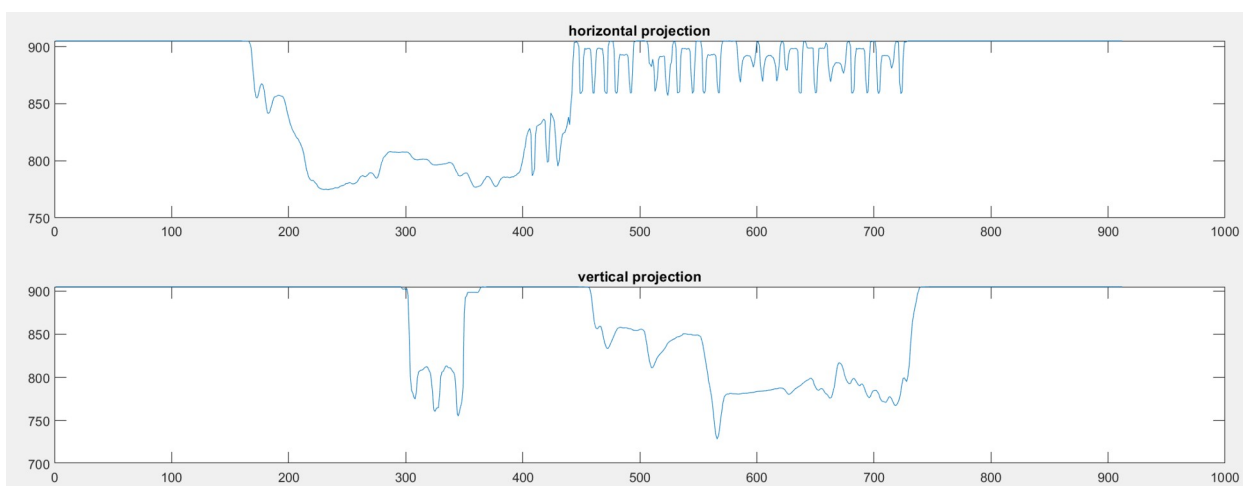
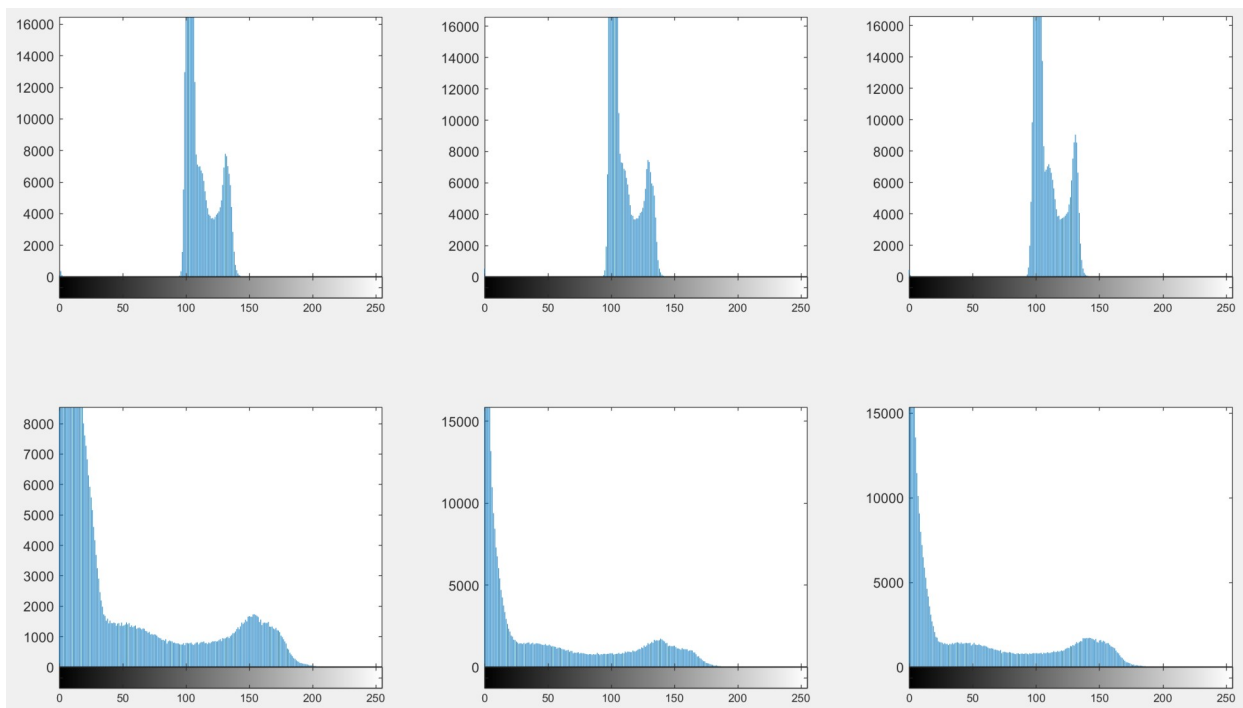
Изобразил профиль изображения текста, а также дополнительно — штрихкода. По профилю заметен пробел в тексте, а в штрихкоде — толстые и тонкие линии.

2.d Результирующие изображения

Изображения — результаты работы в Части 1:

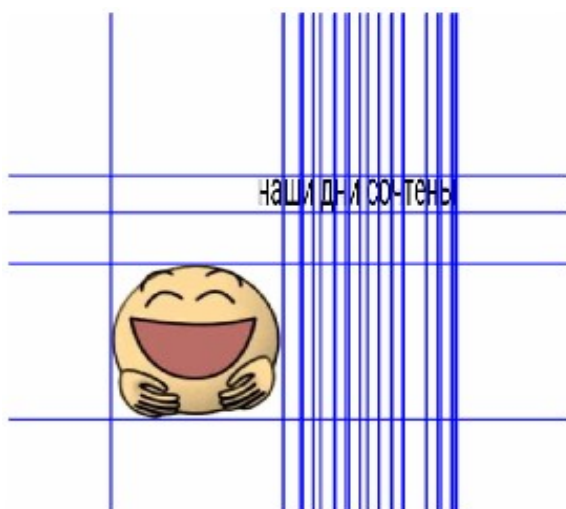


Изображения — гистограммы из Части 2 пункта 1:

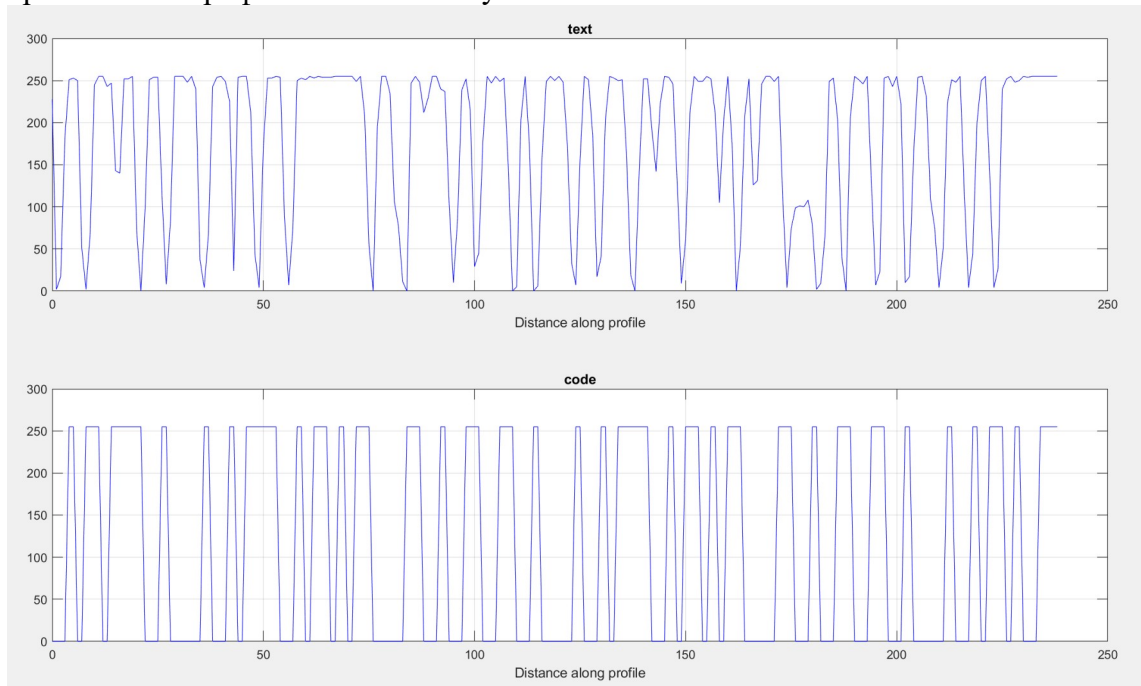


Изображения — проекции из Части 2 пункта 2:

Изображение — границы объектов, определенные по проекциям с помощью пороговых значений:



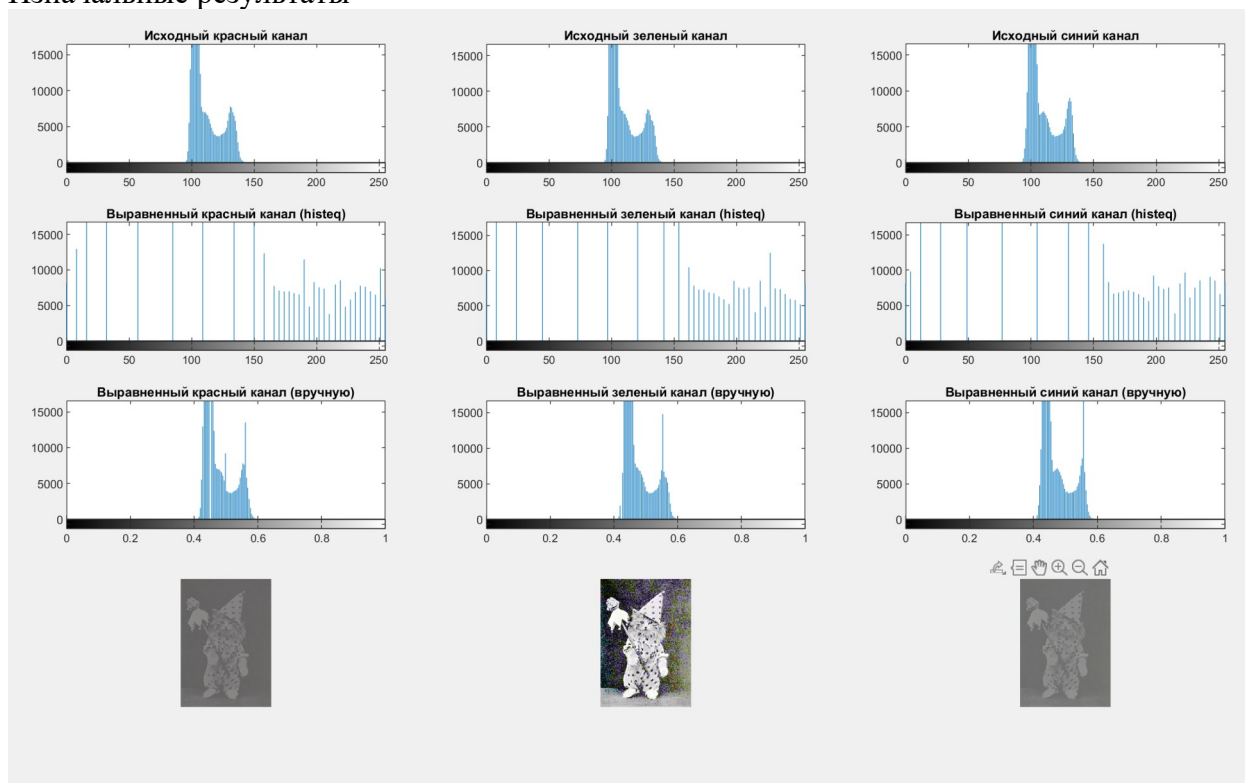
Изображения — профили из Части 2 пункта 3:



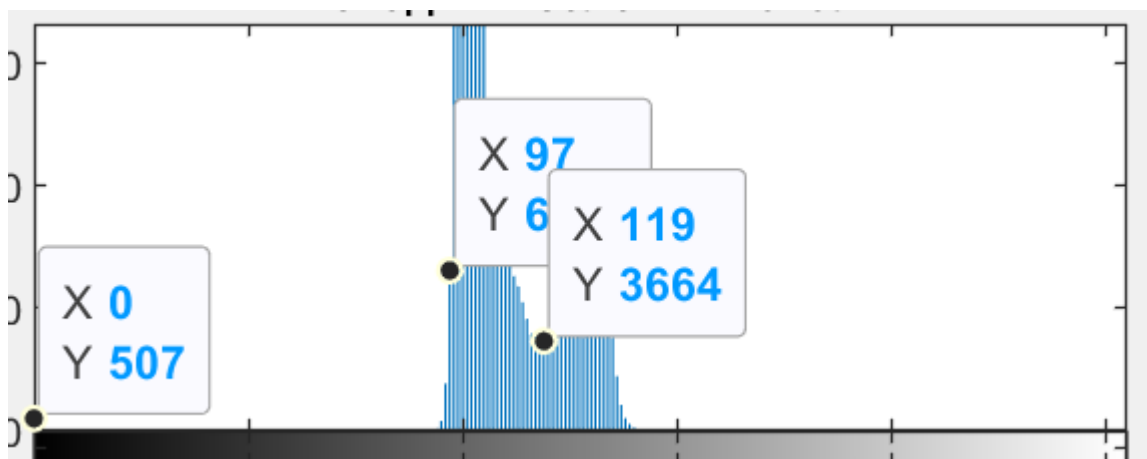
4.

Дополнительное задание — выравнивание гистограммы вручную и с помощью функций MATLAB

Изначальные результаты

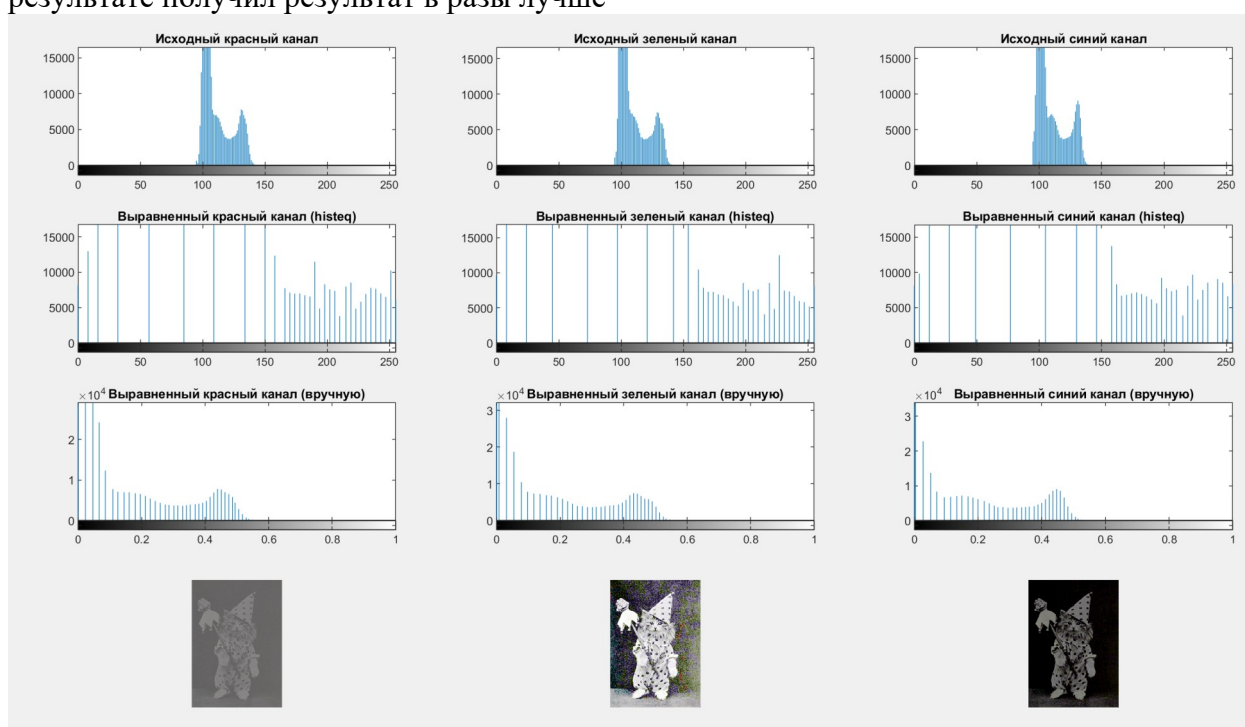


В качестве выравнивания гистограммы вручную использовался способ «Растяжение динамического диапазона». Однако он не сильно преобразовался. Почему? Основная причина — то, что в изображении все-таки присутствуют пиксели, находящиеся далеко от основного скопления «серости».

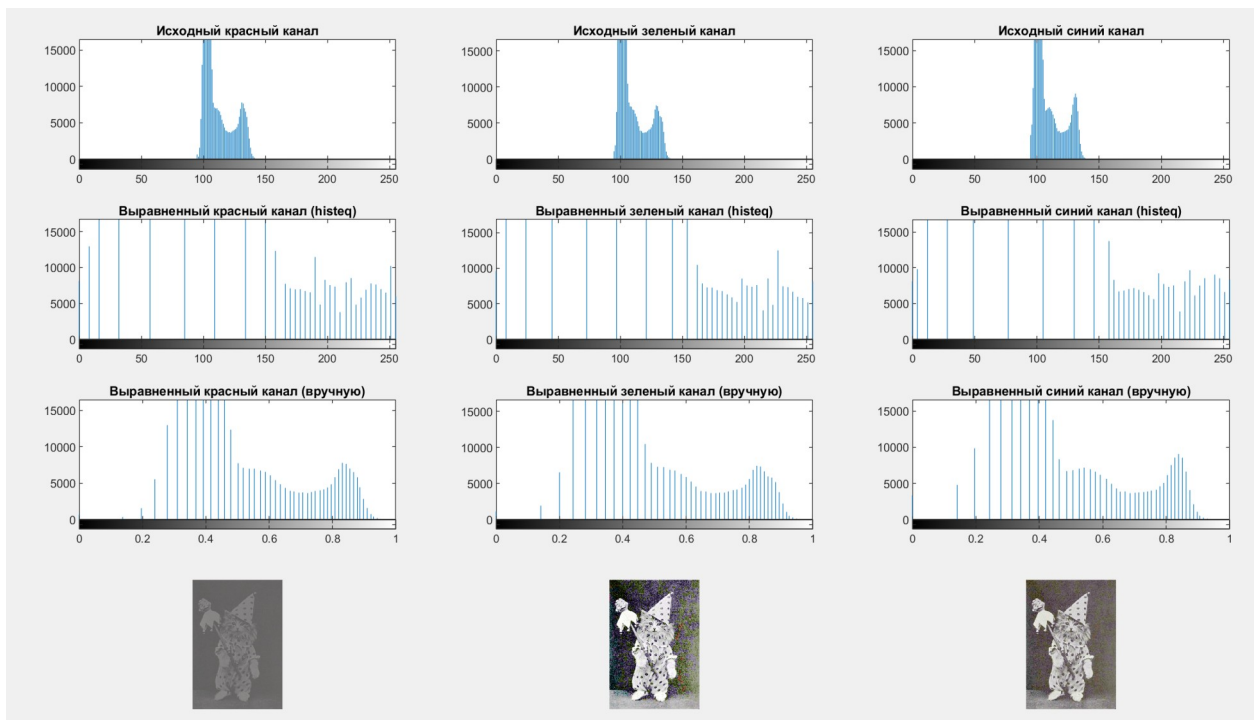


На изображении — самая левая плашка.

Я решил убрать все пиксели яркостью меньше 95 (левого края большого скопления). И в результате получил результат в разы лучше



Также я использовал способ выравнивания арифметической операции для гистограмм до исправления — так как изображение получалось слишком светлым. После исправления я убрал это преобразование — так как теперь изображение получалось слишком темным:



4. Вывод по работе

В результате проведенной работы получилось:

В Части 1: после работы с разными цветовыми слоями изображения, преобразованием их, в результате получил как полутоновые изображения, так и выделения определенных цветов. В конце получилось вывести бинарные изображения — только черные и только белые пиксели.

В Части 2:

Пункте 1: В результате выведения гистограмм (для сравнения добавил более контрастное изображение). По гистограммам видно, что у контрастного изображения больше вариативность пикселей.

Пункт 2: На проекциях четко видно выделяющиеся объекты, из чего мы делаем вывод, что по проекциям можно определять границы объектов (что я в итоге и сделал с помощью пороговых значений).

Пункт 3: В результате видим и можем сделать вывод, что по профилям можно увидеть как линии в штрихкоде, так и пробелы в тексте.

Дополнительное задание:

Выравнивания гистограмм позволяют преобразовывать изображения в более четкие. В результате сравнения встроенной функции и преобразования вручную, видно, что встроенная функция выделяет цвета сильнее, чем написанная мной.