

```

clear;
clc;
I = imread('Untitled 01 1.jpg');
%=====улучшение исходного изображения номера и его бинаризация=====
I0=im2double(I);          %привести к диапазону яркостей 0...1
figure, imshow(I0),title('Origin'); %и вывести на экран fig_1
h = fspecial('gaussian',30,10); %создать маску фильтра Гаусса для
I1 = imfilter(I0,h,'replicate'); %выделения фона и выполнить размытие
figure, imshow(I1,[],)title('Background'); %вывести на экран фон fig_3
I2 = imsubtract(I0,I1);      %вычесть из исходного изображ. фон
figure, imshow(I2),title('Without background');%и вывести на экран fig_4
I2max=max(I2(:));           %вычислить максимальное значение яркости
I3=imadjust(I2, [0 I2max],[],1); %для повышение контраста изображения
figure, imshow(I3),title('After preprocessing');%и вывод результата на экран
%=====бинаризация предобработанного изображения=====
TR=graythresh(I3);         %вычислить порог бинаризации по
                             %минимуму гистограммы
lbw1 = im2bw(I3,TR);        %бинаризация с использованием
                             %вычисленного порога
figure, imshow(lbw1,[],) , title(['Binari TR =',num2str(TR) ]);%вывод
                             %на экран
se=strel('square', 3);      %создать структурообразующий элемент 3x3
lbw2=imclose(lbw1,se);     %выполнить морфологическое замыкание
figure, imshow (lbw2);      %и вывести изобр. на экран
[M n]=bwlabel(lbw2,8); %найти восьмисвязные пиксели на бинарном изображении
                             %M -размеченная матрица, n -число найденных объектов
figure,imshow(M);title(['n =',num2str(n)]);%вывод размеченной матрицы
L=label2rgb(M);             %раскрашиваем найденные области
figure,imshow(L);title('Regions'); %и выводим на экран
%=====поиск номерной пластины=====
%найдем ldata - массив структур с полями, которые описывают различные
%числовые характеристики областей в бинарном изображении M, которое было
%получено функцией BWLABEL
ldata= regionprops(M, 'area','boundingbox','Image','ConvexArea');
%Area-площади объектов.
%BoundingBox-вектор 1x4, определяющий наименьший прямоугольник
                %вокруг области [x y x_width y_width]
%Image-двоичное изображение,с размерами соответствующими BoundingBox.
%ConvexArea-площадь,с размерами BoundingBox,с заполненными дырами.

k = 1;
for i=1:length(ldata)        %цикл с числом повторений равный числу
                             %найденных областей
    ratio = ldata(i).BoundingBox(3)/ldata(i).BoundingBox(4);%вычислить
                             %отношение ширины к высоте
                             %найденных объектов.
    if ldata(i).ConvexArea > 2500 && ratio < 5 && ratio > 3%если площадь
        %объекта больше 2500 и отношение ширины к высоте лежит в диапазоне
        %3...5, то это наш и для него вычисляем ограничивающий

```

```

%прямоугольник bound
bound = floor(Idata(i).BoundingBox);%ограничивающий прямоугольник
        %отвечающий условиям
Im=(Idata(i).Image);    %двоичное изображение с размерами
        %как у BoundingBox
numbers(k) = Idata(i);
%numbers(k).Image = imcrop(M,bound);
k = k +1;
end
end
Iplate=imcrop(I2,bound); %вырезаем из исходного изображения найденный
figure, imshow(Iplate, []),title(['Iplate']); %фрагмент с номерной пластиной
        %и выводим на экран
figure, imshow(Im, []),title(['BWplate']);%fig_9 выводим двоичное изображение
        %с размерами как у BoundingBox
imwrite(Im,'BWplate.bmp'); %сохранить бинарное изоб.номерной пластины в файл
imwrite(Iplate,'Iplate.bmp');%сохранить изображение номерной пластины в файл
%=====

```