

به نام خدا

پروژه ی 2

محدرضا علوی - 810100253

بخش اول:

3- برای تابع طبق رابطه ی گفته شده هر پیکسل را محاسبه می کنیم و ماتریس سه بعدی تبدیل به یک بعدی می شود.

4- برای تابع گفته شده یک آستانه دلخواه انتخاب می کنیم اگر هر پیکسل اگر کمتر بود 0 و اگر بیشتر بود 1 قرار می دهیم.

5- ابتدا تعداد پیکسل های هر جسم را به دست می آوریم برای این کار یک بیت 1 پیدا می کنیم و بیت های اطراف آن را هم پیدا می کنیم و این کار را برای بیت های پیدا شده تکرار می کنیم اگر تعداد پیکسل ها کمتر از مقدار داده شده بود آن جسم را حذف می کنیم.

6- مانند قسمت قبل جسم ها را پیدا می کنیم سپس بیت های هر جسم را از 1 تا تعداد جسم ها قرار می دهیم.

7- هر جسم را با تصویر های دیتا مپ کورولیشن می گیریم و هر کدام که بیشتر بود را انتخاب می کنیم.

تصاویر:

```
1 - clc;|
2 - clear;
3 - close all;
4 - [file, path] = uigetfile ('*.jpg;*.bmp;*.png;*.tif', 'Choose an image');
5 - picture = imread ([path, file]);
6 - picture = imresize (picture, [300 500]);
7 - picture = mygrayfun (picture);
8 - picture = mybinaryfun (picture);
9 - picture = myremovecom (picture, 300);
10 - background = myremovecom (picture, 3000);
11 - picture = picture - background;
12 - [l, n] = mysegmentation (picture);
13 - load TRAININGSET;
14 - letters_num = size (TRAIN, 2);
15 - final_result = [];
16 -
17 - for i = 1:n
18 -     [r, c] = find (l == i);
19 -     y = picture (min (r):max(r), min (c):max(c));
20 -     y = imresize (y, [42, 24]);
21 -     corr_nums = zeros (1, letters_num);
22 -     for j = 1:letters_num
23 -         corr_nums(j) = corr2 (TRAIN{1, j}, y);
24 -     end
25 -     [max_corr, index] = max (corr_nums);
26 -     result = cell2mat (TRAIN{2, index});
27 -     final_result = [final_result result];
28 - end
29 -
30 - file = fopen ('number_Plate.txt', 'wt');
31 - fprintf (file, '%s\n', final_result);
32 - fclose (file);
33 - winopen('number_Plate.txt')
```

تصویر 1 - اسکرپت p1

```

1 function picture2 = mygrayfun (picture)
2     picture2 = 0.299 * picture (:, :, 1) + 0.578 * picture (:, :, 2) + 0.114 * picture (:, :, 3);
3 end

```

تصویر 2 – تابع mygrayfun

```

1 function picture2 = mybinaryfun (picture)|
2     for i = 1:300
3         for j = 1:500
4             if (picture(i,j) < 100)
5                 picture2(i, j) = 1;
6             else
7                 picture2(i, j) = 0;
8             end
9         end
10    end
11 end

```

تصویر 3 – تابع mybinaryfun

```

1 function picture = myremovecom (picture, n)|
2     objects = find_objects (picture);
3
4     for i = 1:length (objects)
5         object = objects{i};
6
7         if (length (object) < n)
8             picture(object(1, :), object(2, :)) = 0;
9         end
10    end
11 end

```

تصویر 4 – تابع myremovecom

```

1  function [l, n] = mysegmentation (picture)
2      objects = find_objects (picture);
3      n = length (objects);
4      l = zeros (size (picture));
5
6      for i = 1:n
7          object = objects{i};
8          l(object(1, :), object(2, :)) = i;
9      end
10 end

```

تصویر 5 – تابع mysegmentation

```

1  function objects = find_objects (picture)
2      [r, c] = find (picture == 1);
3      points = [r'; c'];
4      objects_count = 1;
5      points_count = size (points, 2);
6      objects = {};
7
8      while (points_count > 0)
9          first_point = points(:, 1);
10         points(:, 1) = [];
11         [points, near_points] = find_near_points (points, first_point);
12         object = [first_point near_points];
13         near_points_count = size (near_points, 2);
14
15         while (near_points_count > 0)
16             near_points2 = [];
17
18             for i = 1:near_points_count
19                 [points, near_points3] = find_near_points (points, near_points(:, i));
20                 near_points2 = [near_points2 near_points3];
21             end
22
23             object = [object near_points2];
24             near_points = near_points2;
25             near_points_count = size (near_points, 2);
26         end
27
28         objects{objects_count} = object;
29         objects_count = objects_count + 1;
30         points_count = size (points, 2);
31     end
32 end

```

تصویر 6 – تابع find_objects

```

1 function [points, near_points] = find_near_points (points, first_point)
2     different = abs (points - first_point);
3     index = find (different (1, :) <= 1 & different (2, :) <= 1);
4     near_points = points (:, index);
5     points(:, index) = [];
6 end

```

تصویر 7 – تابع find_near_points

بخش دوم:

مانند یخش قبل با مپ ست پلاک های فارسی انجام می دهیم.

بخش سوم:

روش اول: برای پیدا کردن پلاک قسمت آبی سمت چپ پلاک را پیدا می کنیم. برای این کار ابتدا قسمت های آبی را پیدا می کنیم با دستور `imbainarize` و انتخاب ترشولد مناسب قسمت مورد نظر را پیدا می کنیم سپس قسمت های کوچک و نویز را حذف می کنیم با داشتن نسبت ابعاد پلاک و قسمت سمت چپ پلاک کل پلاک را پیدا می کنیم و جدا می کنیم سپس مانند قسمت قبل عمل می کنیم.

روش دوم: قسمت آبی سمت چپ پلاک را با تصویر کورولیشن می گیریم و شیفتمی دهیم و این کار را تکرار می کنیم. در نقطه ای که کورولیشن ماکسیمم است سمت چپ پلاک است با توجه به نسبت ابعاد پلاک کل پلاک را انتخاب می کنیم و از تصویر جدا می کنیم سپس مانند قسمت قبل عمل می کنیم.