

# Reading and Processing Plaque Images with MatLab

## Contributors

1. Ali Padyav - 810199388
2. Pouriya Tajmehrabani - 810199395

## Part 1 (English)

### Creating the dataset

```
TRAIN = cell(2, len);

for i = 1:len
    TRAIN(1, i) = {imread(['Map Set', '\', cell2mat(nam(i))])};
    temp = cell2mat(nam(i));
    TRAIN(2, i) = {temp(1)};
end
```

### Getting file

```
[file, path] = uigetfile({'*.jpg;*.bmp;*.png;*.tif'}, 'Choose an image');
s = [path, file];
picture = imread(s);
picture = imresize(picture, [300 500]);
```

Ask us to chose an image and resize it to 300\*500 pixels for matrice calculation.

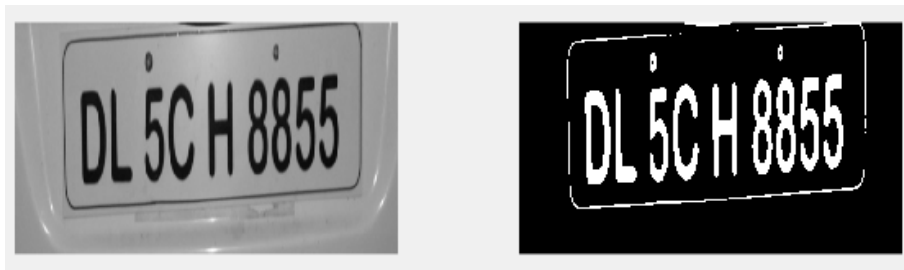


### Preparing image

```
%RGB2GRAY
picture=rgb2gray(picture);
figure
subplot(1,2,1)
imshow(picture)

% THRESHOLDING and CONVERSION TO A BINARY IMAGE
threshold = graythresh(picture);
picture = ~im2bw(picture,threshold);
subplot(1,2,2)
imshow(picture)
```

We convert rgb to grayscale to make a bitmap of image by threshold.

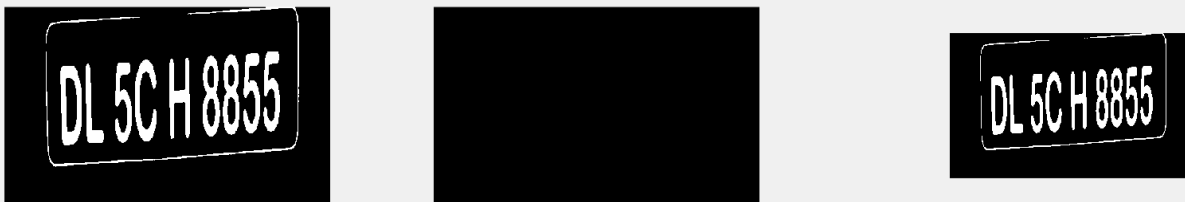


## Cleaning

```
picture = bwareaopen(picture, 200); %removes all connected components
                                     %that have fewer than 30 pixels

background = bwareaopen(picture, 3000);
picture2 = picture - background; %removes all connected components
                                   %that have more than 30000 pixels (background)
```

Then we filter image noise and background.



## Labeling

```
[L, Ne] = bwlabel(picture2);
propied = regionprops(L, 'BoundingBox');
```

By bwlabel we find components and determine it's size and position by regionprops.



## Decision Making

```
for n = 1:Ne
    [r, c] = find(L == n);
    Y = picture2(min(r):max(r), min(c):max(c));
    Y = imresize(Y, [42, 24]);

    ro = zeros(1, totalLetters);

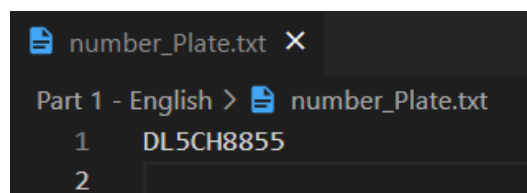
    for k = 1:totalLetters
        ro(k) = corr2(TRAIN{1, k}, Y);
    end

    [MAXRO, pos] = max(ro);

    if MAXRO > .45
        out = cell2mat(TRAIN{2, pos});
        final_output = [final_output out];
    end
end
```

We resize components to specific size to calculate correlation of it and train of letters.

The letter that has most correlation will be chosen and written in final\_output.



## Part 2 (Farsi)

### Creating the dataset



By using these bitmaps, we make train just like part1.

### Main image recognition

#### Find blue parts of plate

```
diff_im = imsubtract(rgb_pic(:, :, 3), rgb2gray(rgb_pic));  
diff_im = medfilt2(diff_im, [3 3]);  
diff_im = im2bw(diff_im, 0.15);  
diff_im = bwareaopen(diff_im, 100);  
  
[labeledImage, numberOfObjects] = bwlabel(diff_im);
```

Use a median filter to filter out noise

check if the piece contains blue in the rgb original picture to make sure it's a plaque



## Determine registration part of plate

```
for n = 1:size(blobMeasurements, 1)
    blobMeasurements(n).BoundingBox(1) = blobMeasurements(n).BoundingBox(1) + blobMeasurements(n).BoundingBox(3);
    blobMeasurements(n).BoundingBox(2) = blobMeasurements(n).BoundingBox(2) - 15;
    blobMeasurements(n).BoundingBox(3) = blobMeasurements(n).BoundingBox(3) * 10.2;
    blobMeasurements(n).BoundingBox(4) = blobMeasurements(n).BoundingBox(4) + 15;
end
```

Because image size is invariant, we determine registration part by blue part.

For certainty extend part a little more.

## Cleaning

```
cropped_piece = imcrop(rgb_pic, blobMeasurements(n).BoundingBox);

car_plaque = rgb2gray(cropped_piece);
threshold = graythresh(car_plaque);
car_plaque = ~im2bw(car_plaque, threshold);
car_plaque = car_plaque - bwareaopen(car_plaque, 2000);
car_plaque = bwareaopen(car_plaque, 30);
```

Removing noise and background just like part 1.

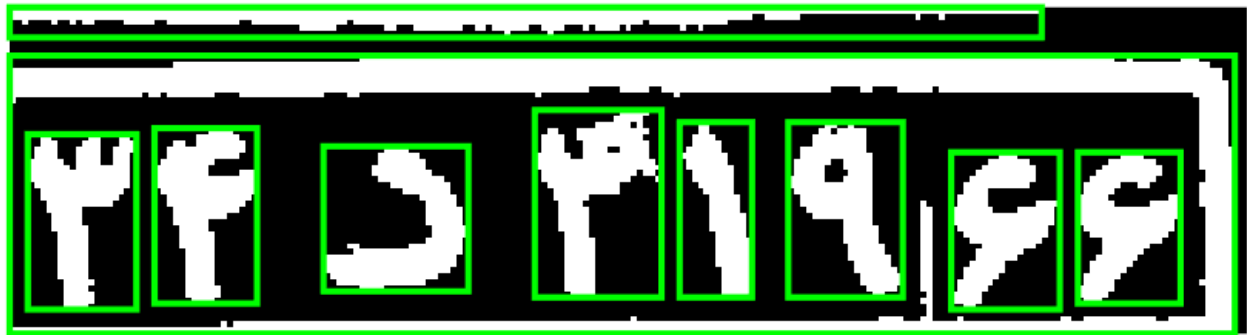


## Labeling

```
bw = bwlabel(diff_im, 8);
stats = regionprops(bw, 'BoundingBox', 'Centroid');
size_of_stats = length(stats);

if (size_of_stats > 0)
    ...
    ...
```

If this registration part has components, we continue rcognition.



## Evaluation

```
for j = 1:Ne
    [r, c] = find(L == j);
    Y = car_plaque(min(r):max(r), min(c):max(c));

    m = size(Y, 2);
    k = size(Y, 1);
    aspect_ratio = m / k; %aspect ratio of letters
    if (aspect_ratio >= 3) || (aspect_ratio <= 1/3)
        continue
    end

    Y = imresize(Y, [70, 50]);
    ro = zeros(1, totalLetters);

    for k = 1:totalLetters
        ro(k) = corr2(TRAIN{1, k}, Y);
```

```

end

[MAXRO, pos] = max(ro);
if MAXRO > .45
    out = cell2mat(TRAIN(2, pos));
    final_output = [final_output out];
end

end

```

If component aspect ratio isn't in letters ratio interval, we skip it, otherwise correlate it with letters train and find the best choice (like part 1).


## Saving

```

file = fopen('number_Plate.txt', 'wt');
fprintf(file, '%s\n', final_output);
fclose(file);

```

At last save final\_output to file.

 final\_output 'T1۹۶۶۵۳۴'

Disarrangment is because of farsi letters.

<https://github.com/SyntheticDemon/Car-Plaque-Image-Processing/>