

## Overview

In this document I'll provide the steps used in the process, along with the reasons behind the decisions taken.

## 1. Data Acquisition

### 1.1. Source Choice

First, we tried to find a dataset for gray-scale license plates' images. However, found none. Then, we tried to find gray-scale license plates' images, but found none as well. Then, we accepted colored images, and found some. All the image in *testData* folder are found by *Google* searching.

Unfortunately, we couldn't find all the letters/digits easily, so we tried an approach that produced really good results. Our approach is to use artificial data for training that has all the digits, and then test on real data (the images found by *Google* search). This approach actually produced ~70% accuracy using blocked-centroid classifier, and ~92% accuracy using the classifier that had tracks and sectors. Our source of artificial data was [this webpage](#). All data in *trainData* folder is artificial data gotten from the mentioned webpage.

### 1.2. Scope

This is the specifications of input data that our classifier works on:

- We accept colored images.
- The image must be the license plate, not a generic scene.
- We recognize English capital letters and English data. And we believe that due to our artificial source, we could extend that easily to special characters and small letter characters. But finding license plates to test with could be cumbersome.
- License plate can have a variable number of characters.

## 2. Preprocessing

### 2.1. Scaling and coloring

First, we scale the image to 100x200 resolution. Then, we convert it to black and white. We use *im2bw* for the color conversion. For our algorithm to classify well, the character needs to have different color from the background after color conversion. We have no problem what color the background is and what color the foreground is, we assume that the background will have greater number of pixels, and

according to so, we make the character's color white and the background's color black.

## 2.2. Image Segmentation/Character Extraction

We turn the problem into an *OCR* problem by extracting the characters from the license-plate image. We use our own approach for so.

Our algorithm is as follows:

*Detect 8-connected white components in the image*

*For each component:*

*Consider it to be a rectangle and calculate its width and height*

*If its dimensions are significant enough, consider it to be a character*

*Otherwise, discard it*

For the parameters we use to check the significance: We calculated them by trying different parameters and seeing if it'll extract characters in the training data correctly (I'm not sure, but for honesty, we may have used the test data as well. In the beginning, they were mixed).

## 3. Classification

We use 2 algorithms for classification: *Blocked Centroid*, and *Contour-detection* (the one was sectors and tracks). The accuracy for *Blocked Centroid* was ~70%, and for *Contour-detection* was ~92%. Exact method of error calculation can be found in *calcError.m*.