Final Project Report: CV part

TITLE: A multi-processor based Vehicle Retrieval System

Course: ECE 7220- Real time embedded computing

Instructor: Dr.DeSouza

Ning, Guanghan   pawprint:gnxr9

1. Introduction

License Plate Detection and Recognition is a traditional computer vision task. In the project, it is achieved with fairly good result and some novelties. This report is describing the Vehicle Retrieval System computer-vision-wise.

2. Target Problem:

The target problem is to detect from a static image(the speed of detection is improved so that it is also fast enough for video clips) the license plate of a vehicle. Once the plate is detected, alignment of the plate is made to ensure the following segmentation and character recognition render good results. Ultimately, the license plate number of the vehicle is derived from the image.

3. Implementation of License Plate Detection and Character Recognition:

3.1 License Plate Extraction

Preparation: Extract HOG features from the training images of license plates. Use SVM to train the model. 3378 license plates are trained in all.

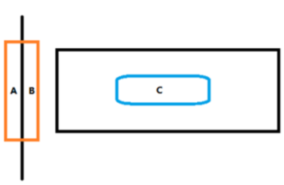
Apply edge detection on the input image first. Scan the whole image with a scanning window, and the regions with edges above a certain ratio will perform SVM classification in the window to decide if this window's score is above the threshold and therefore positive, meaning it contains a plate. So the scanning is not over the whole image, which saves time for detection. The HOG features for the whole image are computed beforehand, instead of calculating that with each scan, so it saves time too. The image will be resized to 909*523, and the scanning window is multiple scales by setting the resized image multiple scales. So for each scale the HOG features will be re-calculated. The nearby positive windows will merge to get the final location using k-means.

(**Fig1**. Edge image(left) and Scanned region(right))

3.2 Alignment

(Fig2. The extracted plate before and after Alignment**)**

(Fig3. The ROI before and after alignment)



(**Fig4.** The model for alignment)

As figure4 shows, for a license plate, a dynamic line will be located as the aligned edge, where the fitness score is maximized. The fitness score comes from:

score_left= dist_RARC- weight_left*dist_RCRB

score_right= dist_RARC- weight_right*dist_RCRB

score_up= dist_RARC- weight_up*dist_RCRB

score_down= dist_RARC- weight_down*dist_RCRB + dist_RARB;

dist_RARC is the HSV histogram distance between region A and Region C. Other notations will be likewise.

The idea is to make use of the color information to align. The region A should be very different from B and C, and region B should be very similar to region C, in terms of color.

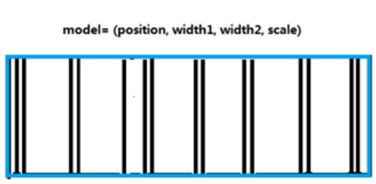
3.3 Segmentation

Based on accurate alignment, k-means is used to derive two clusters from the license plate. One is the characters, and one is the background of the plate. Set the major cluster with more points white and the minor cluster black so the characters will be black. Segmentation is based on the binary license plate.

(**Fig5**. Aligned binary license plate)

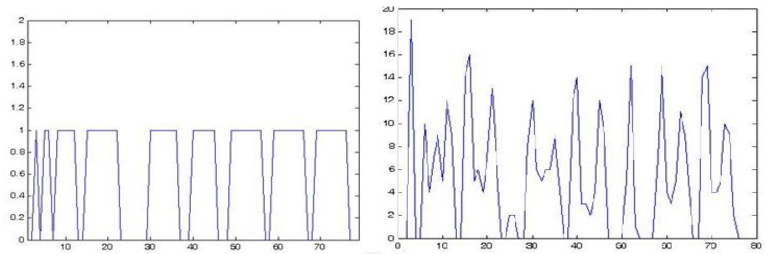
(**Fig6**. Segmented license plate)

The segmentation is done with a model. The model is shown in Fig7. It is from prior knowledge that we know the number of characters from Chinese license plates is seven, and that there is a dot between the second and the third character.



(**Fig7**. Segmentation model)

The model is deformable with its position and width and scale. Scan vertically at each x axis of the plate, the 255 to 0 inversion contributes to the score. In fig8, the y axis shows the score at the corresponding x axis of the plate.



(**Fig8**. the analysis of the best fit model. left figure is with threshold; right figure is without threshold.)

3.4 Character Recognition

Character recognition is a 1 vs N classification problem. SVM is used here to classify multiple times until classification result is positive. The extracted features that are used for character recognition are:

Number of black to white changes vertically and horizontally, black/white ratio horizontally and vertically, and the raw feature.

There are 5 set of features in all. They are concatenated and the overall feature length is: 96.

3.5 Dataset and Results:

The plate detection is tested on two datasets, one is daylight and contains 43 images, the other one is night and contains 62 images.

The detection rate is both 100% and there is no false positives. All of them are well aligned.

For night dataset, the speed for detection is around 200ms on average with 3 scanning scales and the scanning step width being 8. For the daylight dataset, since the plates are fairly small, in order to ensure 100% detection rate, the scales are still 3, but the step width to scan is set down to 2. And the average detection time is around 1000ms. If we tolerate some miss, the time could be around 200ms too.

The Segmentation is 100% correct.

400 characters are trained, and another 100 are tested. The testing characters are derived from the segmentation output patches. The character recognition rate for segmented single characters is 99%. (The demo is attached in the final project bundle)