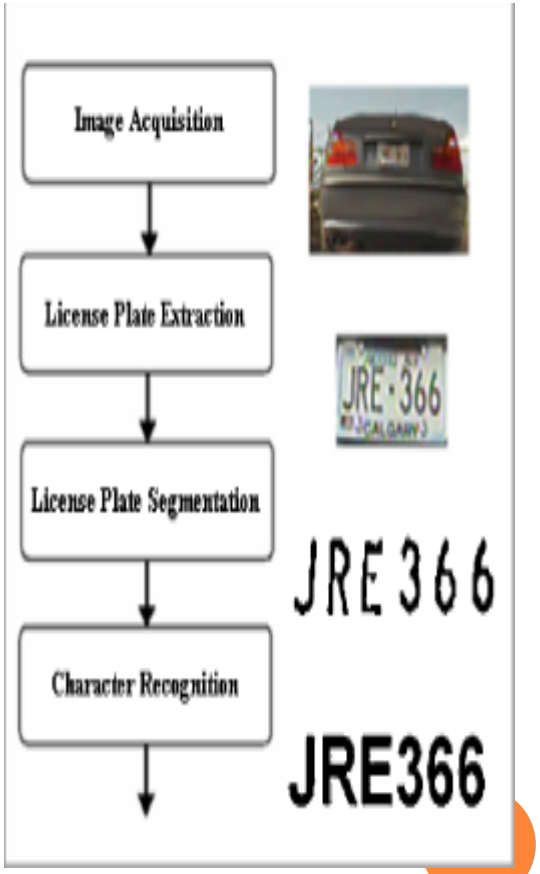
Automatic License Plate Recognition(ALPR) on embedded system

Presented by Guanghan

APPLICATIONS

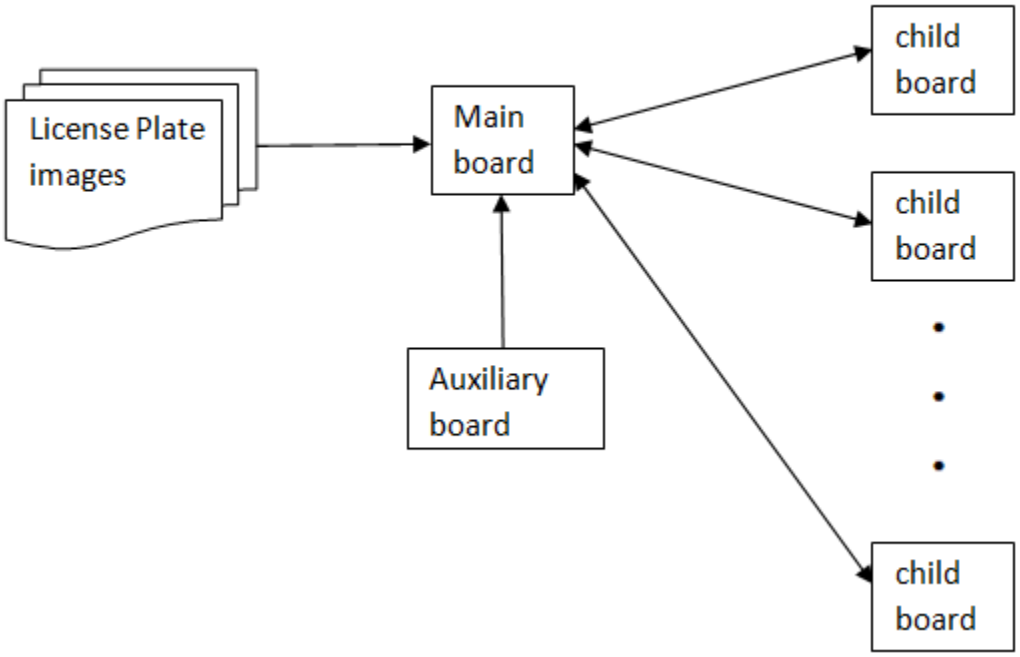
1. Automatic toll collection
2. Traffic law enforcement
3. Parking lot access control
4. Road traffic monitoring

ALPR SYSTEM: SEVERAL STAGES



My research in the real-time embedded system project

Hardware part:

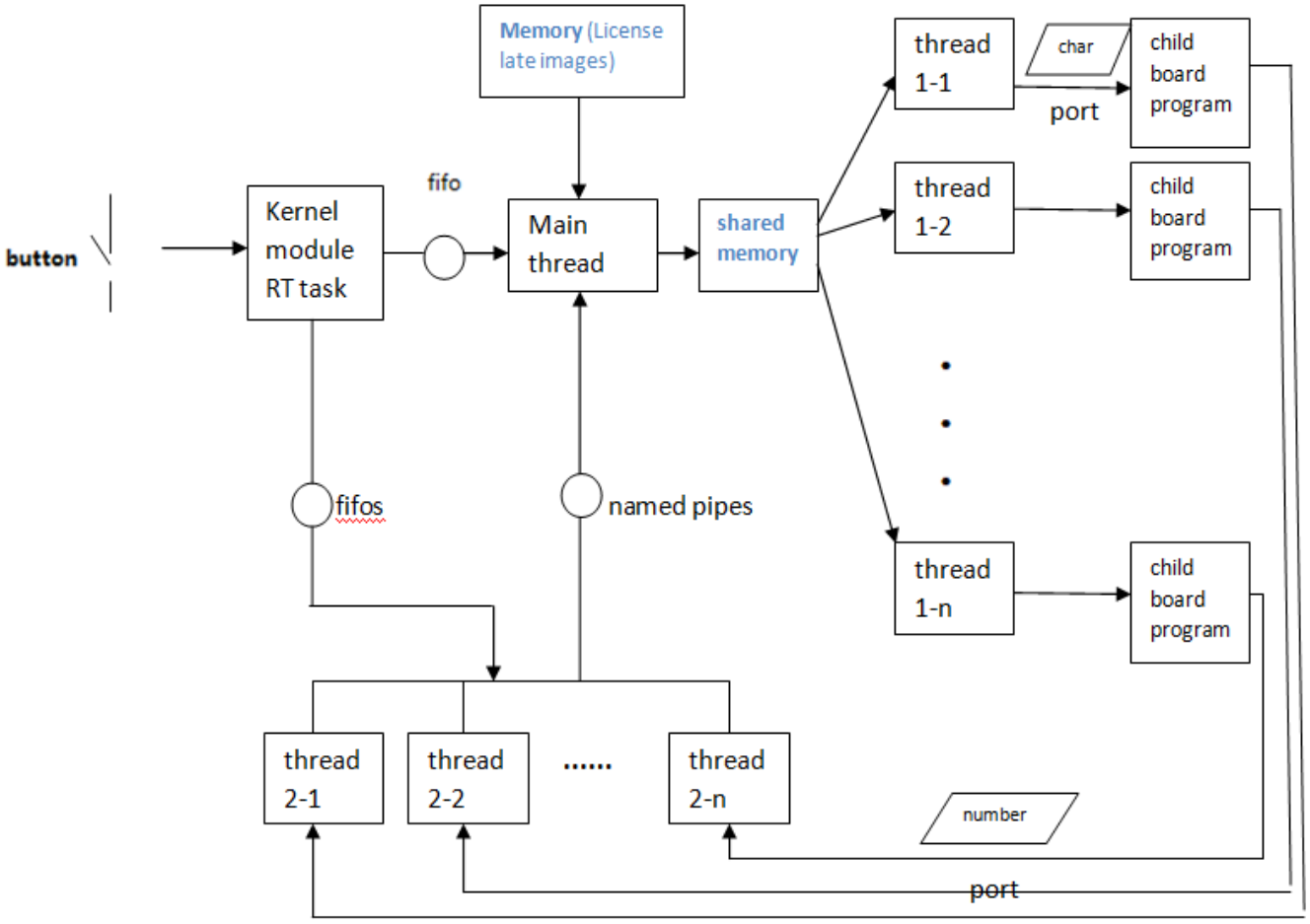


My research in the real-time embedded system project

- The "main board" performs segmentation on license plate images. Its auxiliary board provides time to capture/load license plate images.
- "Child boards" need to receive segmented characters from the "main board". The characters are transmitted through **ports**.
- The "child boards" will recognize the character received and then send the recognized number back to the "main board" through ports.
- The "main board" exhibits the recognized numbers.

My research in the real-time embedded system project

Software part: main board

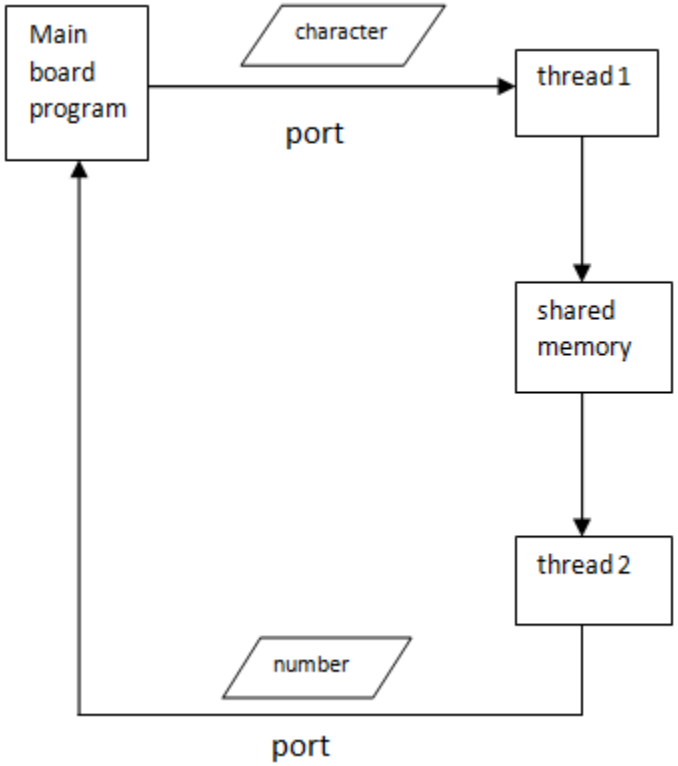


My research in the real-time embedded system project

- **The "main board" side**
- **The module** for the "main board" will check the B0 button of its auxiliary board.
- Once B0 button on the auxiliary board of the TS-7250 "main board" is pressed, which simulates the process of this Vehicle retrieval system capturing the image of vehicles, the module will send the message through a **fifo** to the main thread, who will then read license plates from a pre-defined directory. The main thread will thereafter perform segmentation of the plate.
- The **user space program** for the "main board" have **three set of threads**. Aside from the main thread mentioned above, one set of threads will ask from the main thread for the segmented characters through a **shared buffer**, then send the segmented characters to "child" boards via the ports.
- Another set of threads will do the job of receiving recognized numbers from "child boards". Once the numbers are received, the threads send them to the main thread through several **named pipes**.
- Once the main thread receives the recognized numbers, since it knows which number comes from which thread, it then prints out the numbers in the **same order** as they are shown on the plate. **Semaphores** will be used as a form of synchronization. The main thread only prints out the results after the threads that send the numbers are synchronized.

My research in a real-time embedded system project

Software part: child board



My research in a real-time embedded system project

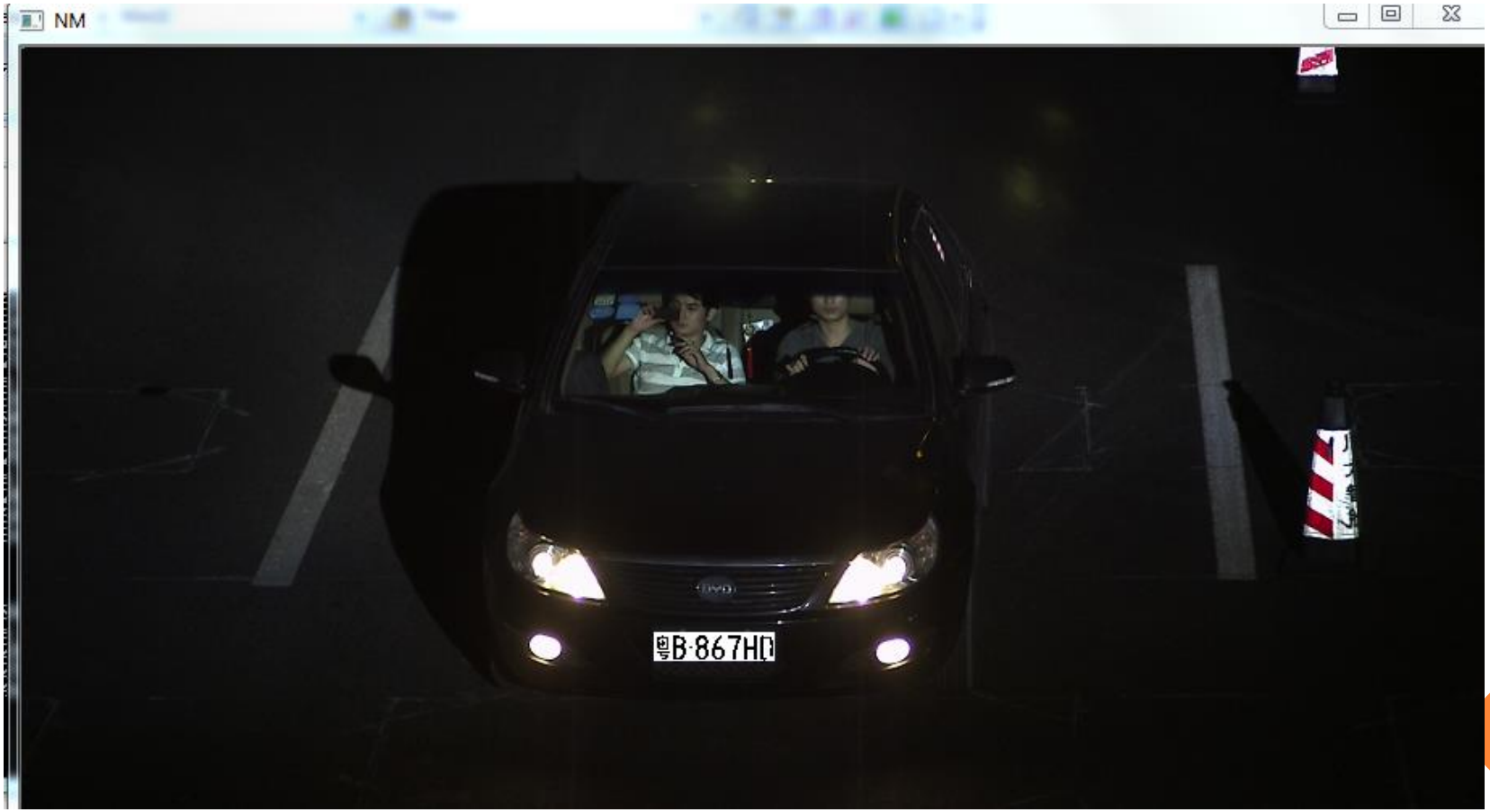
- **The "child boards" side**
- There are **two threads** for the **user space program**(for each of the child board). For each child board, one thread receives the corresponding character from the "main board", put them in a **shared buffer**. The other thread reads from the shared buffer the segmented character, perform character recognition, and then send the recognized number back to its corresponding "main board"-thread.

My research

- 1. License Plate extraction
- Scan the whole image.
- Canny edge detection to find possible regions.
- Extract HOG features from these possible regions.
- SVM to classify whether one block contains license plate.
- K-means to merge nearby possible regions.

My research

- 1. License Plate extraction

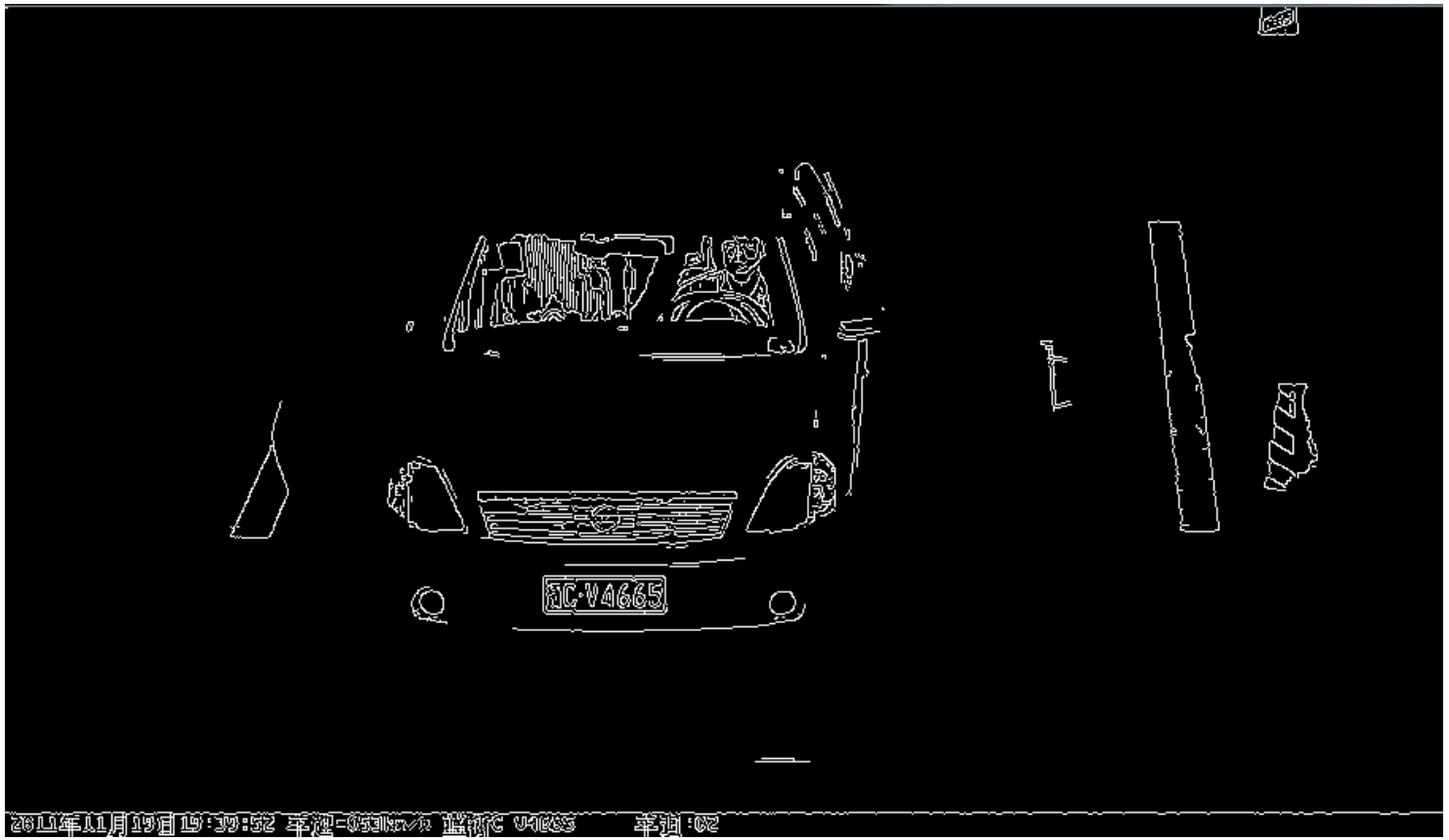


My research

- 1. Licence Plate Extraction
- Use edge information. Only scan regions with edges. (Cut time half.)
- Use correlation between scales. Information transfer to the next scale through a size-deformable cascade mask.(Cut another half)

My research

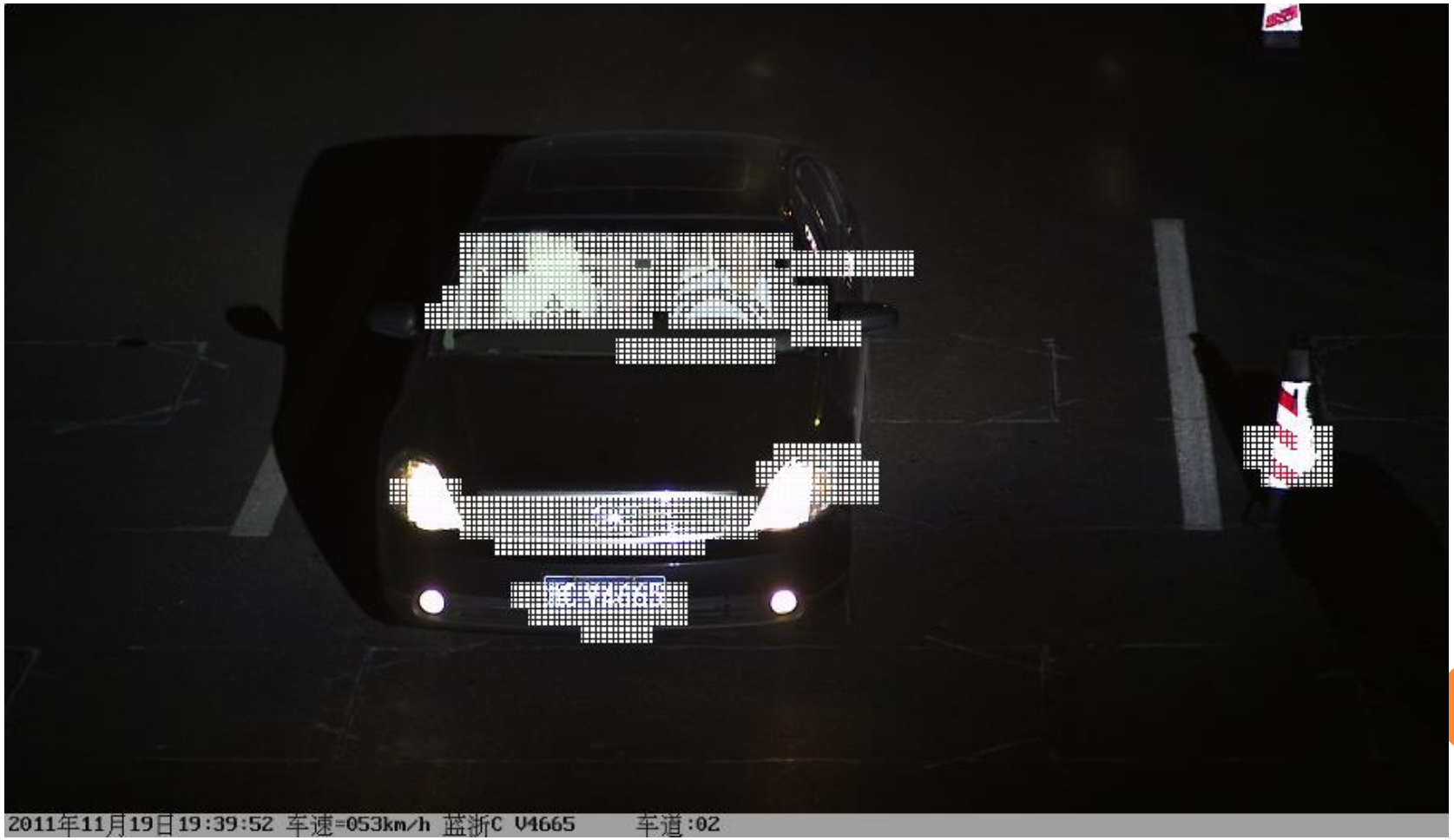
- License Plate Extraction
- Regions that are covered by the scan. Rectangle is plate-size. Will Merge.



MY RESEARCH

1. License Plate Extraction

Scan region: Large amount of time is saved. 250ms for two scales.



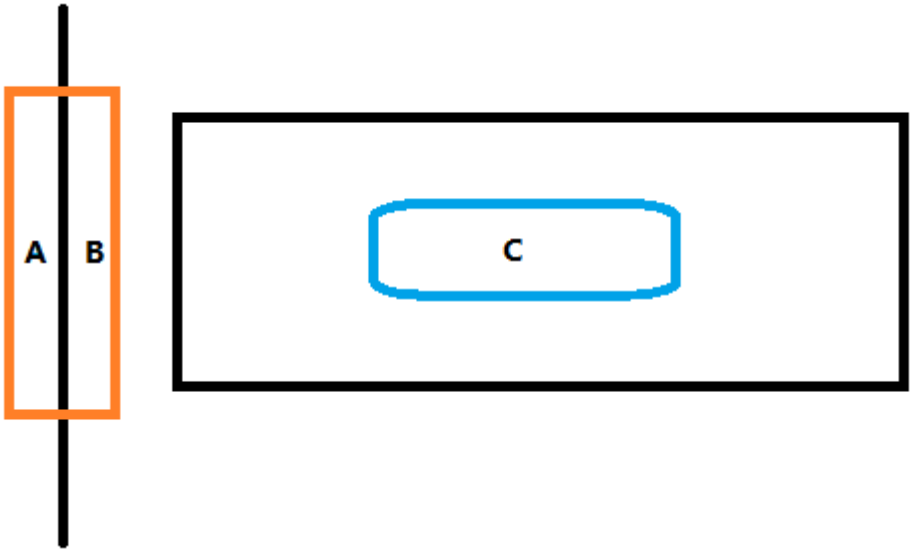
My research

- 2. Alignment
- Align the four lines of the plate so that the ROI fits the plate contour very well.
- Use Color information.

My research
2. Alignment

My research
2. Alignment

RGB hitogram distance



score= dist_RARC- weight_left*dist_RCRB
score= dist_RARC- weight_right*dist_RCRB
score= dist_RARC- weight_up*dist_RCRB

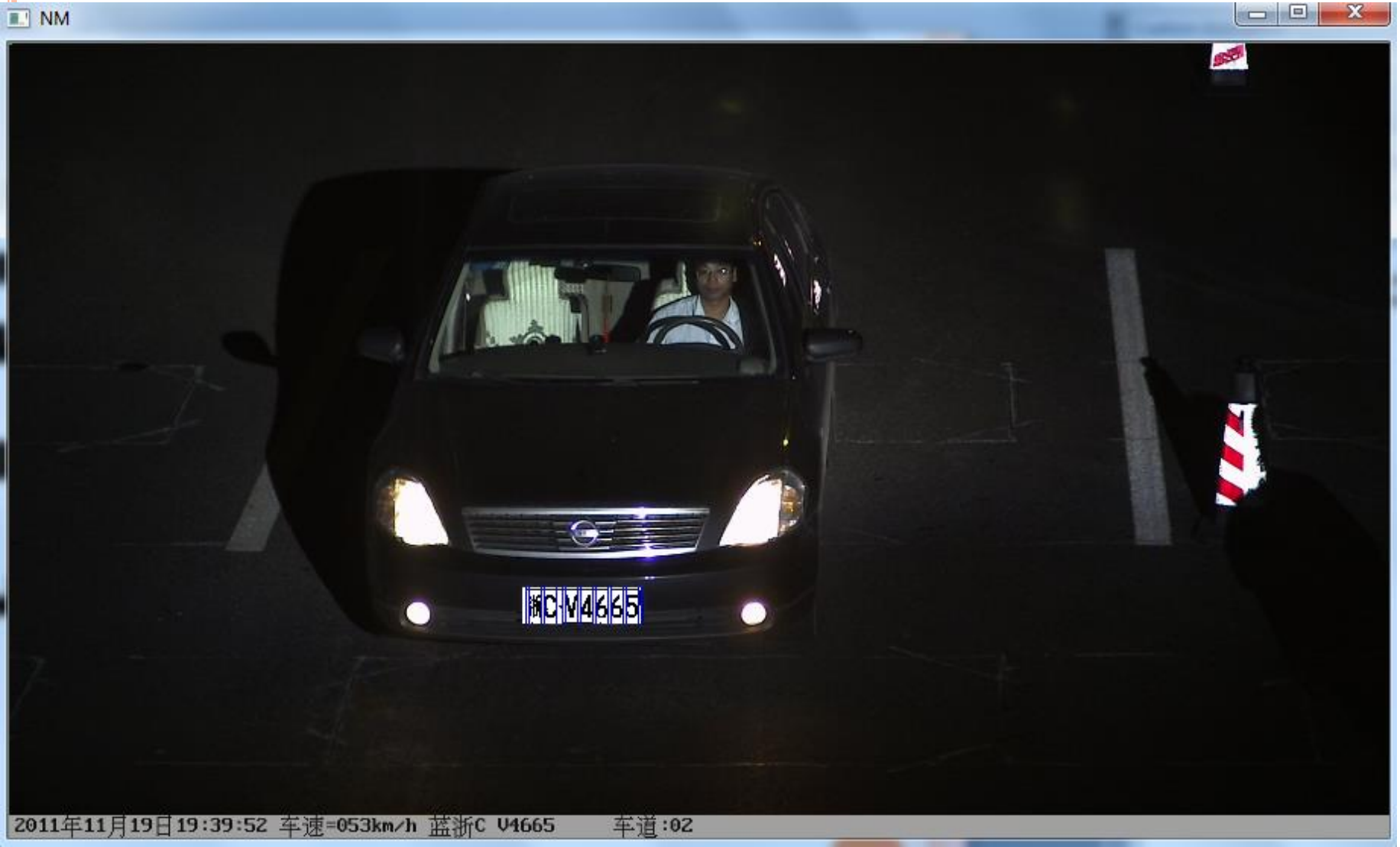
score= dist_RARC- weight_down*dist_RCRB + dist_RARB;

My research

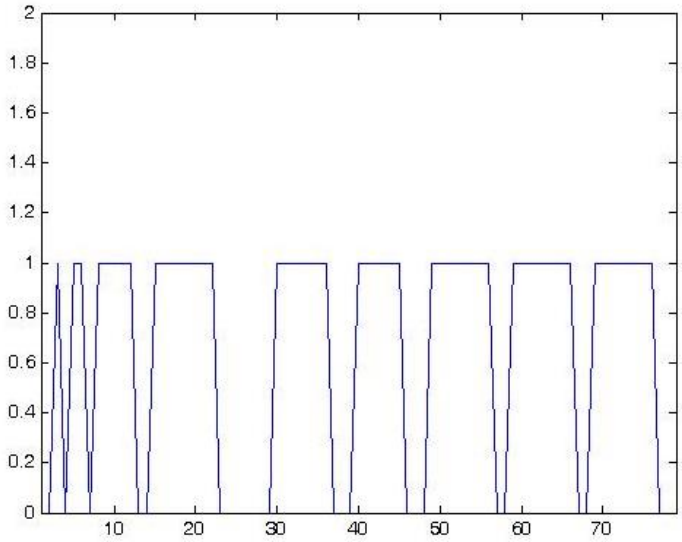
- 3. Segmentation
- Use prior knowledge: Know how many characters exist in a license plate.
- A corresponding model.

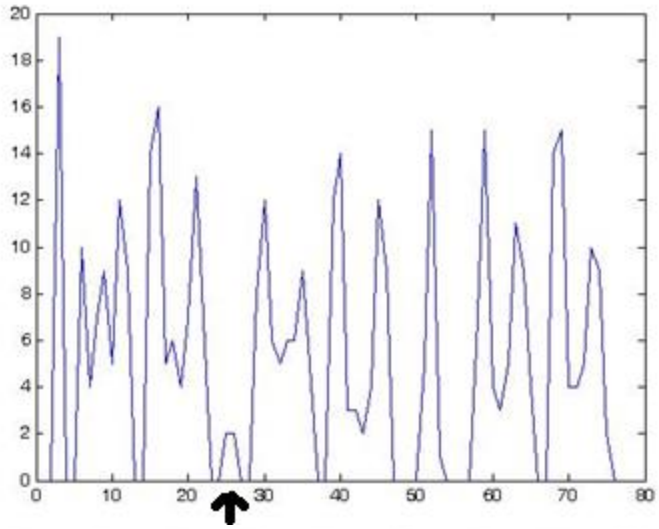
- Use k-means to binarize the aligned plate.
- Optimize the score.
- Scan vertically. In model's character region if black score adds, in space region if black score decreases.

My research



My research
2. segmentation





Scan vertically.
0 to 255 change: ct++

Threshold to binary

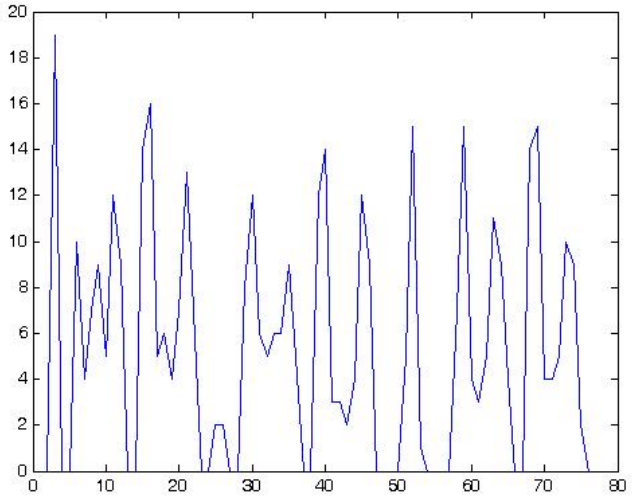
No threshold

My research
2. segmentation

- Prior Knowledge

Segmentation model

model= (position, width1, width2, scale)



My research

- 4. Recognition
- Features:
- 0 to 1 changes in each column
- 0 to 1 changes in each row
- 1 ratio in each column
- 1 ratio in each row
- Raw feature

- Classifier:
- SVM

My research Results:

- Based on 62 test images.
- Detection rate: 100%
- Segmentation: 100%
- Recognition: 99%