1. **License plate**
   1. **Standard format**
      1. **Private vehicle**

(#)## L ####

* 2 initial digits:
  + 01 -> 69: typical passenger vehicles
  + 70 -> 79: vans, recreational vehicles, coaches
  + 80 -> 97: freight vehicles (truck) (\*)
  + 98 -> 99: specialized vehicles (e.g., tow trucks)
* 3 initial digits:
  + 100 -> 699: typical passenger vehicles
  + 700 -> 799: vans, recreational vehicles, coaches
  + 800 -> 979: freight vehicles (truck)
  + 980 -> 999: specialized vehicles (e.g., tow trucks)
* Letter:
  + Private vehicles: 가, 나, 다, 라, 마, 거, 너, 더, 러, 머, 버, 서, 어, 저, 고, 노, 도, 로, 모, 보, 소, 오, 조, 구, 누, 두, 루, 무, 부, 수, 우, 주
  + Rental cars: 허, 하,호
* 4 final digits: 1000 to 9999, assigned separately regardless of car type
  + 1. **Commercial vehicle**

[Province/City]##

L ####

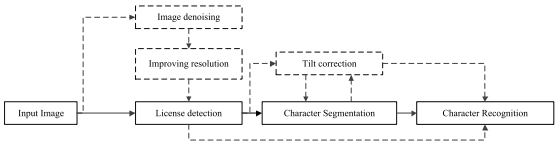
* Province/City: province/city name written full in Hangeul, minus 도 or 시 suffix: 강원, 경기, 경남, 광주, 대구, 대전, 부산, 서울, 세종, 울산, 인천, 전남, 전북, 제주, 충남, 충북
* 2 initial digits:
  + 11 -> 69: taxi
  + 70 -> 79: vans, buses
  + 80 -> 97: freight vehicles (truck) (\*)
  + 98 -> 99: specialized vehicles (e.g., recreational vehicles)
* Letter:
  + Taxi/bus: 바, 사, 아, 자
  + Rental cars: 허, 하,호
* 4 final digits: 1000 to 9999, assigned separately regardless of car type
  1. **Difficulties**

**License plate deflection:** license plate tilts (vertical tilt, horizontal tilt, or both) may cause character distortion and adversely affect character recognition.

**Image with noise:** in the real scene, the license plate image will distort suffered by rain line, snow line and other noises, and some license plates might be defaced.

**Fuzzy license:** license plate usually occupies a smaller proportion in the image, which might cause challenges due to their low resolution and noisy representation.

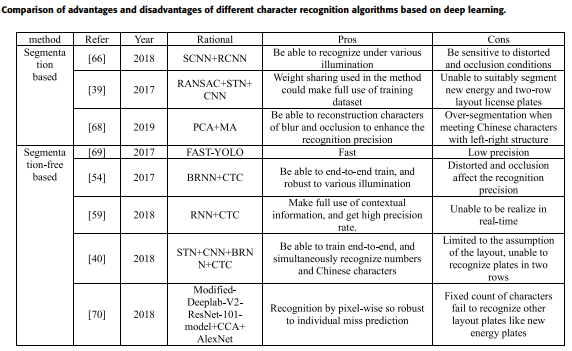
* 1. **License plate recognition process**



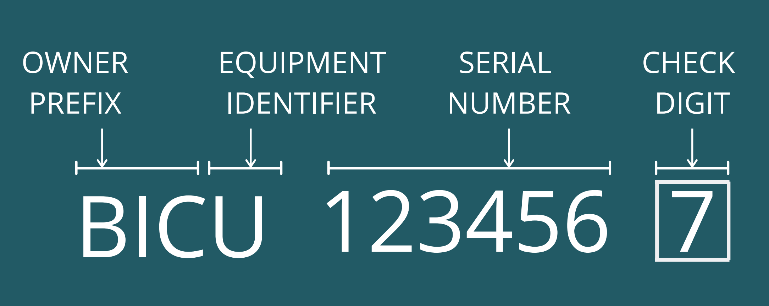
* 1. **Existing algorithms comparison**

The algorithms based on the **segmentation** divide the vehicle license plate into single characters

The algorithms based on the **segmentation-free** transform the vehicle license plate recognition problem into character sequence labeling



1. **Container code**
   1. **Standard format (ISO-6346)**



* Owner prefix: first three capital letters
* Equipment category identifier:
  + U: freight container
  + J: detachable freight container
  + Z: trailers and chassis
* Serial/registration number: decided by the owner
* Check digit: to determine whether the entire code is valid or not
  1. **Check digit computation**
* Step 1: an equivalent numerical value is assigned to each letter in alphabet (beginning with 10 for the letter A, but skip number 11)
* Step 2: each of the numbers calculated in step 1 is multiplied by where is the exponent to base 2 and pos starts at 0, from left to right
* Step 3:
  + Step 3.1: summarize up all results of step 2
  + Step 3.2: divide the result by 11
  + Step 3.3: round the result down towards 0
  + Step 3.4: multiply the integer value by 11
  + Step 3.5: subtract result of (3.4) from result of (3.2), the result is check digit

|  |  |  |
| --- | --- | --- |
|  | **License plate** | **Container code** |
| **Localization** | YOLO (v4, tinyv3, Fast, L)  Multi-task CNN  Bi-LSTM  Transcription  WPOD-Net  Clustering (density-based spatial, k-Means++)  Vision-based techniques  Hybrid cascade  Template matching | ResNet + UNet  Faster R-CNN  EAST  CTDBNet  Bi-LSTM  Text line connection  MSER  Connected Domain Analysis  CTPN  HED |
| **Segmentation** | CRNet | PSENet  Gradient descent projection  Histogram projection |
| **Recognition** | CNN  OCR-Net | CRNN  CTPN  Backpropagation  CTRNet  LeNet |