

# Multi-Style License Plate Recognition System using K-Nearest Neighbors

**Soungsill Park\*, Hyoseok Yoon and Seho Park**

Contents Convergence Research Center, Korea Electronics Technology Institute  
Seoul, South Korea

[e-mail: sspark@keti.re.kr, hyoon@keti.re.kr, sehohpark@keti.re.kr]

\*Corresponding author: Seho Park

*Received September 26, 2018; revised March 12, 2019; accepted April 14, 2019;  
published May 31, 2019*

---

## Abstract

There are various styles of license plates for different countries and use cases that require style-specific methods. In this paper, we propose and illustrate a multi-style license plate recognition system. The proposed system performs a series of processes for license plate candidates detection, structure classification, character segmentation and character recognition, respectively. Specifically, we introduce a license plate structure classification process to identify its style that precedes character segmentation and recognition processes. We use a K-Nearest Neighbors algorithm with pre-training steps to recognize numbers and characters on multi-style license plates. To show feasibility of our multi-style license plate recognition system, we evaluate our system for multi-style license plates covering single line, double line, different backgrounds and character colors on Korean and the U.S. license plates. For the evaluation of Korean license plate recognition, we used a 50 minutes long input video that contains 138 vehicles of 6 different license plate styles, where each frame of the video is processed through a series of license plate recognition processes. From two experiments results, we show that various LP styles can be recognized under 50 ms processing time and with over 99% accuracy, and can be extended through additional learning and training steps.

---

**Keywords:** license plate recognition, license plate structure classification, multi-style license plate, k-nearest neighbor

---

A preliminary version of this paper was presented at APIC-IST 2018, June 24-27, Nha Trang, Vietnam. This extended version includes updated materials on a multi-style license plate recognition system, implementation and evaluation results. This work was supported by the IT R&D program of MOTIE/ KEIT. [10077018, Development of integrated embedded platform and commercial prototype equipped with high-speed LPR function]

## 1. Introduction

License plate recognition (LPR) systems are widely used and studied camera-based applications [1]. From 1970 to the present, extensive research on LPR has been developed and deployed to various environments employing several different LPR methods [2][3][4][5][6]. Furthermore, recent advances in camera hardware now allow researchers and engineers to work with higher resolution inputs from wide angles to realize feature-rich automatic LPR systems [7][8][9][10]. Indeed, LPR systems are already adopted in different types of intelligent transportation systems for parking control, traffic monitoring and smart tolling systems and its application areas continue to grow. Still, all LPR systems for different use cases are affected by varying license plate (LP) styles, weather conditions and light exposure of where they are deployed [11][12][13].

Limitations of most LPR systems concern their LPR capabilities for working with various styles present on LPs. We can group most LPs into two common structures or types based on a number of character lines in the LP. So, if an LP contains only one line of characters then it is a single line plate (SLP) structure whereas an LP with two lines of characters is a double line plate (DLP) structure as shown in Fig. 1.



Fig. 1. Various LP styles based on two common structures of SLP and DLP

Fig. 1 shows Korean LPs that have different styles represented by different fonts, background color, character color, character properties (i.e., width, height and spacing). For SLPs, character size (width and height) and spacing are consistent while these properties vary for DLPs. Most countries use an SLP structure while a few countries such as China and South Korea use a DLP structure. For this reason, previous LPR systems focused on recognizing LPs of a strictly limited SLP-based style [8][9]. For a small number of LPR systems that support the DLP structure, they put restrictions on background and character colors of recognizable LPs [9].

In this paper, we propose and evaluate a multi-style LPR system for recognizing various LP styles that differ in background colors, character colors, and character properties, based on both SLP and DLP structures. The proposed LPR system (hereafter, abbreviated as MSLPRS) integrates an LPR pipeline consists of LP detection, LP type classification, character segmentation, and character recognition processes. We employ a machine learning algorithm,

K-Nearest Neighbors (KNN) for character recognition that can be extended to multi-languages through a pre-training of target characters.

## 2. Related Work

A typical previous LPR system operates in four stages that include 1) an initial process for obtaining input data, 2) a follow-up process for LP detection, 3) a subsequent process for character segmentation, and 4) the last process for recognizing the segmented characters. Due to interconnections between LPR processes, the final result is much affected by a preceding LP detection method used. Since an LP may or may not be present in an input image, the next LPR process only executes when an LP is successfully detected from that input image.

There are six known LP detection methods using boundary features [3][4][11][12][13], global image features [5][8], texture features [9][10], color features [14], character features [2][6], and two or more combined features [7], respectively. Using these methods, LPR systems detect LPs by finding a rectangle in the input [3][4] or by exploiting LP-related information such as connected objects [5][8], colors [14] and characters [2][6].

After successful LP detection, each character is segmented from the detected LP using pixel connectivity [7], projection profiles [8][15], prior knowledge of characters [2][4], character outlines [13], and combined features [9], respectively. To segment characters, LPR systems consider the degree of rotation [9] and the brightness of the detected LP [1]. Segmented characters are then recognized by pixel comparison or by feature extraction method. Pixel comparison approaches can recognize characters fast, but they are susceptible to noise [3][16]. On the other hand, feature extraction approaches are computation heavy, yet they are robust against noise [7][8][9][17][18][19].

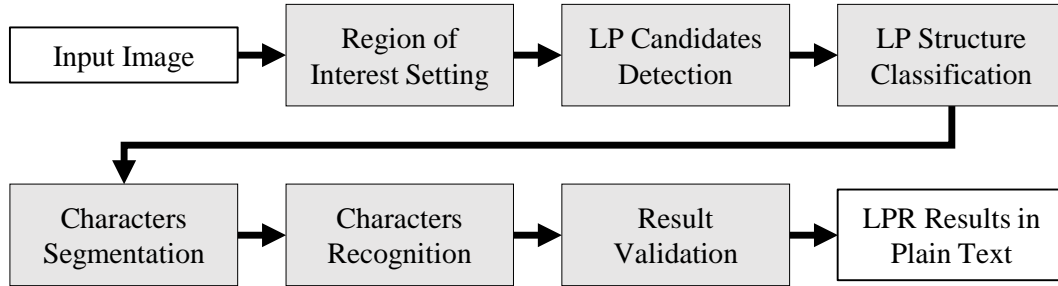
Previous LPR systems have put restrictions on certain LP styles that can be recognized, since it was difficult for LPR systems to detect and recognize multi-style LPs through their preprocessors and LPR pipelines [11][12][13]. To support multi-style LP recognition, our MSLPRS differs from previous systems by introducing two preprocessors and multi-language support through additional learning. **Table 1** shows a summary of previous LPR systems and our proposed MSLPRS.

**Table 1.** A summary of LPR systems and their characteristics (LPD: LP Detection / SC: Structure Classification, CS: Character Segmentation, CR: Character Recognition)

System (Year)	LP Target Country	LPR Methods	Pros and Cons
Yun et al. (2017) [11]	Korea	LPD: Edge detection after adjusting contrast enhancement	- Works with degraded images - Lower recognition rate on night street images
Tabrizi and Cavus (2016) [12]	Iran	LPD: Edge detection CS: Morphology CR: KNN-SVM	- Performs OCR twice to improve recognition rate - 2 stage learning for KNN/SVM
Sarker et al. (2014) [13]	Korea	LPD: Two-stage cascade AdaBoost	- Robust to illumination changes, fast LPD (56.64 ms) - Performs LPD twice
Deb et al. (2009) [14]	Korea	LPD: Sliding concentric window SC: HSI color space CS: Intensity histogram	- Supports LPs with yellow, green, and white background - Sensitive to AoV, physical appearance, and env. conditions

Proposed MSLPRS	Korea/ the U.S.	LPD: DBSCAN clustering SC: Pairing centroid points CR: KNN	<ul style="list-style-type: none"> <li>- Two preprocessing steps for multi-style LPR pipeline</li> <li>- Multi-language support through additional learning</li> </ul>
-----------------	--------------------	--	--

### 3. Multi-Style License Plate Recognition System



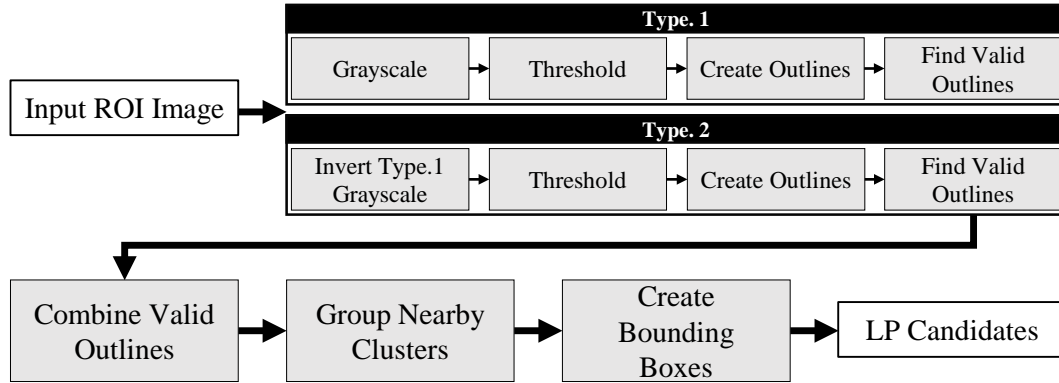
**Fig. 2.** The LPR pipeline for the proposed MSLPRS

**Fig. 2** shows the entire LPR pipeline of the proposed MSLPRS for recognizing multi-style LPs. We designed our MSLPRS and its underlying LPR pipeline to support various environments when a region of interest (ROI) is specified accordingly. At the start of the pipeline, a single image or a video is used as an input to the MSLPRS. The LPR pipeline assumes that camera-fed input images are valid for LP and character recognition (i.e., an LP in input images, if any, is at least  $60 \times 30$  pixels). Then the MSLPRS detects an LP based on the specified ROI and classifies the LP into one of the two pre-defined structures (i.e., SLP or DLP). Since the size and spacing of characters vary based on the detected structure, our MSLPRS reflects this prior knowledge to segment characters on the detected LP efficiently and accurately. To find valid data for character recognition, segmented character parts are combined and noise data is removed. The found valid data is then used for the character recognition process using KNN. The result from the character recognition process is once again validated against the identified LP structure before declaring the final recognition result (i.e., LP number and characters in plain text). In following subsections of this section, we describe each LPR process in detail.

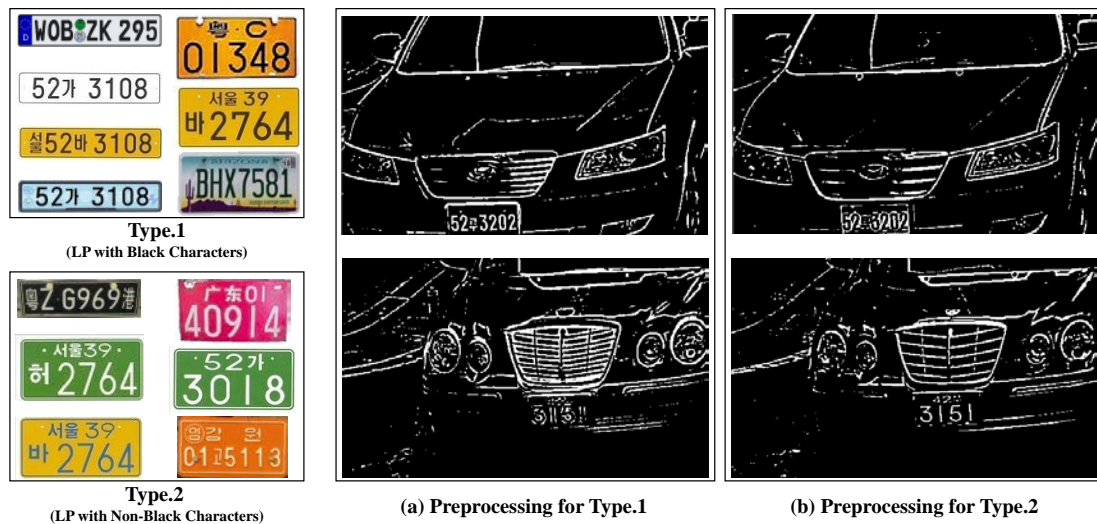
#### 3.1 Region of Interest Setting

There are various environments for LPR systems such as highways, parking lots and streets. Based on where an LPR system is deployed, a driving direction of the vehicles and a style of LPs to monitor change. For example, LPR systems installed on highways recognize multiple LPs from multiple lanes simultaneously. For an LPR system monitoring parking lots, it recognizes a single LP from a single lane at the entrance or at the exit. As these two different use cases illustrate, a coverage or a system's area of interest differs for each LPR system. To recognize LPs efficiently, our MSLPRS uses a pre-specified ROI to recognize LPs within that ROI only. An administrator of the MSLPRS can use one or more ROIs for intended use cases by specifying an ROI with X, Y, Width and Height values. Once the administrator specifies ROIs, the LPR pipeline of the MSLPRS only processes input images within the specified ROI. Therefore, the administrator should carefully select and test ROIs considering their use cases.

### 3.2 License Plate Candidates Detection



**Fig. 3.** A detailed procedure for the LP candidates detection process



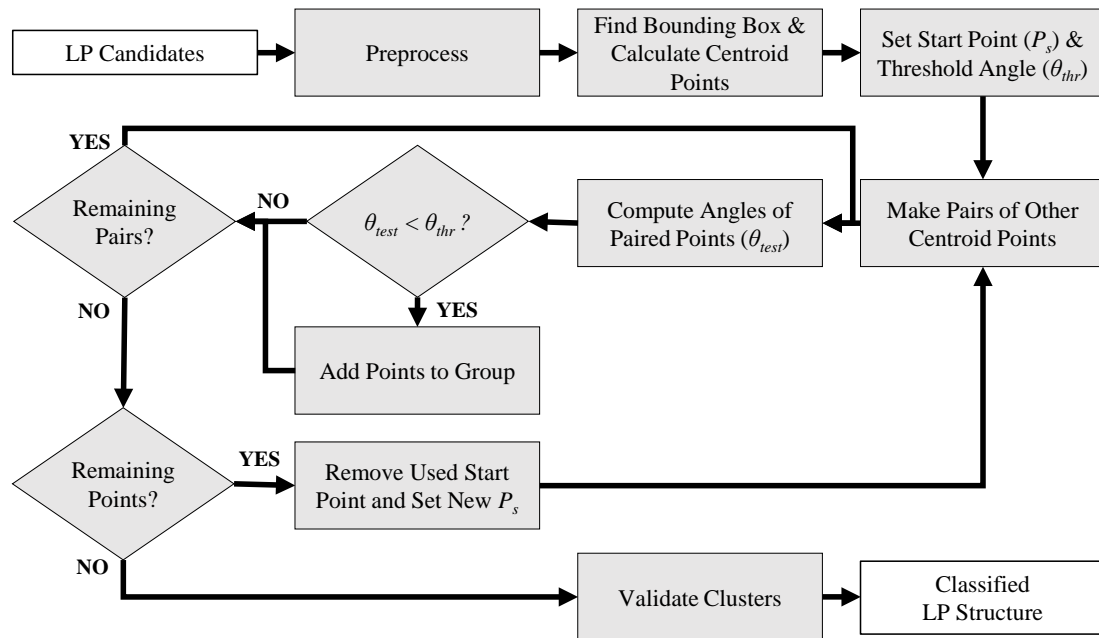
**Fig. 4.** Two pre-processing steps for multi-style LP detection

**Fig. 3** shows a detailed procedure for the LP candidates detection process (LPCDP) using the specified ROI. To support multi-style LPs with different background and character colors, we integrate two types of pre-processing steps to extract valid data from LPs with black characters and non-black characters. First type of pre-processing step is for LPs with black characters that we named Type 1. Second type of pre-processing step is for LPs with non-black characters (i.e., white and blue) that we named Type 2. Both pre-processing steps extract valid outlines from the input image then combines those outlines for further LP detection steps. Valid outlines refer to outlines that exhibit similar properties as characters (i.e., width, height and a ratio of width and height). The LPCDP groups the valid data found into nearby clusters. To group the valid data, we used DBSCAN (Density-based spatial clustering of applications with noise) algorithm [20]. Even though DBSCAN is a data-clustering algorithm that has higher computation load than K-means algorithm, this algorithm is robust to noise and performs better for non-linearly separable clusters. Unlike K-means algorithm that processes

data based on its position, DBSCAN processes data based on its density and shows better performance in noise handling. However, DBSCAN is much slower than K-means algorithm when performing a large number of operations and prone to settings of radius and threshold values used in the algorithm. We found the use of DBSCAN useful for LPR applications, since characters in LP are horizontally and sequentially aligned where valid outlines are close to each other and have similar level of densities. Consequently, DBSCAN can create a bounding box from the valid outlines. Then the LPCDP compares how similar two ratios (i.e., the ratio of a bounding box's two sides with the ratio of a target LP's two sides) are. Up to this point, the LPCDP may detect non-LP objects that are excluded from later processes of the LPR pipeline.

### 3.3 License Plate Structure Classification

The LP structure classification process (LPSCP) classifies LP candidates found from the LP candidates detection process into SLP or DLP structure. To recognize different LP styles based on two structures as illustrated in Fig. 1, the MSLPRS need appropriate and optimized processes to separate two structures. To distinguish one structure from another structure, we use LP-specific style attributes such as size, ratio, and color of target LPs. However, there are limitations and difficulties in distinguishing LP styles with such information. The size and the ratio of LPs vary with different target LP styles. In addition, the color information changes depending on current light exposure. Furthermore, attributes such as spacing, size, and ratio of the characters on the SLP are mostly constant while the same attributes vary in the DLP. Our insight is that we can improve the final recognition results of an LPR system when we introduce an LPSCP to give SLP or DLP structure information before the LP recognition step.



**Fig. 5.** A detailed procedure for the LPSCP

**Fig. 5** shows a detailed procedure for the LPSCP. The input images to the LPSCP go through a preprocessing step that includes resizing, gray image conversion, noise removal, data binarization, outline extraction and bounding box computation. The preprocessing step



first resizes input images to a predefined size of height and width to normalize input images. Then the resized images are converted to gray images. Then Gaussian blur is applied the gray images to remove small noise. Finally, the resulting data are binarized. The binarized data in white and black are used to extract outlines and remove outliers (i.e., whose size or ratio of width and height is too small or too large). As the last step of the preprocessing, valid outlines are used to compute boundary boxes.

Following the preprocessing, centroid points of the bounding box are computed from valid outlines. Then a start point ( $P_s$ ) and a threshold angle ( $\theta_{thr}$  such as  $10^\circ$ ) of the bounding box are specified. Using the specified  $P_s$  as the centroid point, all other points excluding  $P_s$  are paired to each other. If the angle formed ( $\theta_{test}$ ) by two paired points is less than  $\theta_{thr}$ , then they are added to the current group ( $G$ ), otherwise they are removed. Starting with a new  $P_s$ , this procedure of checking  $\theta_{test}$  is repeated for all pairs as points are added to  $G$  or removed. If there is no more data to specify as a  $P_s$ , then the validity of  $G$  is verified.  $G$  is considered valid if it contains 4 or more points. By counting the number of valid  $G$ s, we can distinguish SLP and DLP structures. We experimentally specify and update  $\theta_{thr}$  to distinguish various LP styles. In the case of SLP structure, there is only one valid  $G$  and there are two valid  $G$ s in the case of DLP structure.

Our proposed LPSCP is not a pixel-by-pixel operation. This process is light since it segments LPs by simple operations such as centroid point computation and  $\theta_{thr}$  checks. However, our approach has limitation with an input image that has an LP rotated over  $\theta_{thr}$ , since we only compare data within this threshold. Therefore, rotated input images should be calibrated before the LPSCP. The LPSCP is also used to search for character areas. Since the LPCDP may yield non-character areas, the LPSCP identifies valid character areas to be used in later character segmentation and character recognition processes. As the result of the LPSCP, an area for characters is identified for the SLP structure and two separate lines (i.e., upper and lower lines and character areas are identified for the DLP structure.

### 3.4 Characters Segmentation

Additional steps are required in the process of segmenting characters used in the LP. In the case of Hangul or Korean alphabet, an outline of Korean alphabet composed of the initial (choseong), the medial (jungseong) and the final (jongseong), takes one of the four shapes as shown in Fig. 6.

In the character recognition process (CRP) that follows the character segmentation process (CSP), extracted initial, medial and final are combined, since the CRP recognizes the combined form. Although it is possible to recognize the initial, the medial, and the final separately, the recognition rate based on this approach is low since it is difficult to separate noise data from valid data. To increase character recognition rate, it is necessary to combine consonants and vowels of Hangul. This process of combining characters is necessary for Hangul, but is not necessary for the English alphabet.

The CSP binarizes the input data, extracts outlines and finds valid outlines. Outlines are considered valid if it satisfies two predefined conditions of maximum size and a width/height ratio according to SLP and DLP structures. We determine these conditions based on LP structures and country-specific LP specifications where the LP is used. In most SLPs, character size and proportion are consistent. The character recognition process should consider different DLP styles such as different character size and proportion present in upper and lower lines. Since character spacing of an LP is country-specific, an administrator of MSLPR should adjust ranges of valid outlines according to the deployed environment before

using the MSLPR. By using the administrator-defined variables, the MSLPRS computes valid outlines and eliminates redundant data. To process redundant data, the MSLPRS removes smaller data from the redundant data. This process for removing redundant data is required for a segmented Korean character as shown in Fig. 6. (b) to (d) while characters shown in Fig. 6 (a) does not require this step. The CSP joins separated data within a range that does not exceed the valid outline's maximum width, height and ratio. After the combining process, the CSP removes excluded or remaining small noise data.

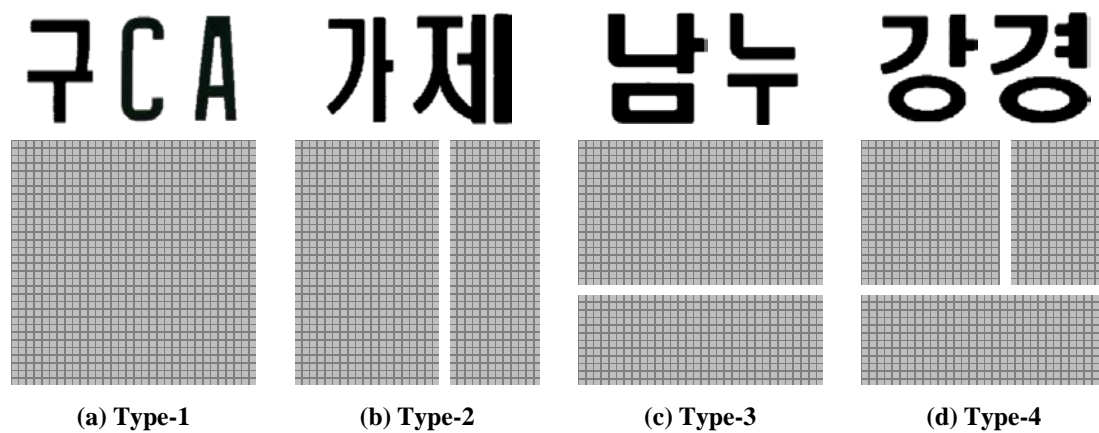


Fig. 6. Four types of LP character composition

### 3.5 Characters Recognition

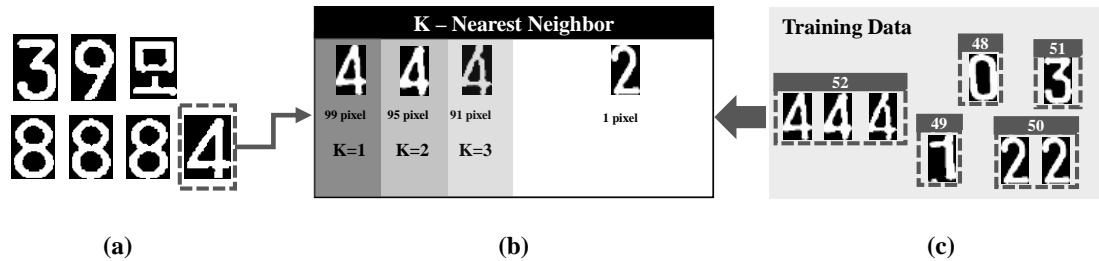
The character recognition process (CRP) uses the valid data found in the CSP as an input to KNN (i.e., a supervised learning method) [21][22]. We train characters on LPs to recognize them in the CRP. Previous optical character recognition studies have provided methods and open software tools to recognize characters from input images. In the case of LPR systems where a limited subset of characters is used, we believe that it is more efficient and accurate to use real LP images for training. To train a data set, we need values for features and labels. Feature values refer to images of target characters to be recognized and label values refer to corresponding Unicode of characters included in the feature values. Through training steps, the MSLPR system learns to recognize characters used in LPs.

Label Values	48	48	48	48	48	48	...	57
Feature Values	Image 1	Image 2	Image 3	Image 4	Image 5	Image 6	...	Image N
Feature Values	<pre> &lt;?xml version="1.0"?&gt; - &lt;opencv_storage&gt;   - &lt;images type_id="opencv-matrix"&gt;     &lt;rows&gt;624&lt;/rows&gt;     &lt;cols&gt;600&lt;/cols&gt;     &lt;dt&gt;f&lt;/dt&gt;     &lt;data&gt; 0. 1. 0. 0. 2. 109. 126. 255. 255. 255. 209.           126. 34. 0. 0. 2. 0. 0. 1. 0. 0. 1. 0. 127. 255.           252. 255. 253. 254. 252. 255. 253. 255. 253.           133. 15. 1. 2. 0. 0. 6. 226. 255. 255. 253.           252. 255. 255. 255. 254. 255. 255. 254. 254. </pre>							
Label Values	<pre> &lt;?xml version="1.0"?&gt; - &lt;opencv_storage&gt;   - &lt;classifications type_id="opencv-matrix"&gt;     &lt;rows&gt;624&lt;/rows&gt;     &lt;cols&gt;1&lt;/cols&gt;     &lt;dt&gt;i&lt;/dt&gt;     &lt;data&gt; 48 48 48 48 48 48 48 48 48 48 48 48 48 48 48           48 48 48 48 48 48 48 48 48 48 48 48 48 48 48           48 48 48 48 48 48 48 48 48 48 48 48 48 48 48           48 48 48 48 48 48 48 48 48 48 48 48 48 48 48           48 48 48 48 48 48 48 48 48 48 48 48 48 48 48 </pre>							

Fig. 7. Preceding training steps for the CRP



**Fig. 7** shows training steps for recognizing characters used in LPs. Input data for the training step are binarized and margin-removed images that contain characters to be recognized as shown in **Fig. 8** (c). Feature values pair with label values (i.e., corresponding Unicode). To be recognizable, all possible characters on LPs need the aforementioned training step. We save the trained data in XML format and use it for the CRP in the MSLPRS.

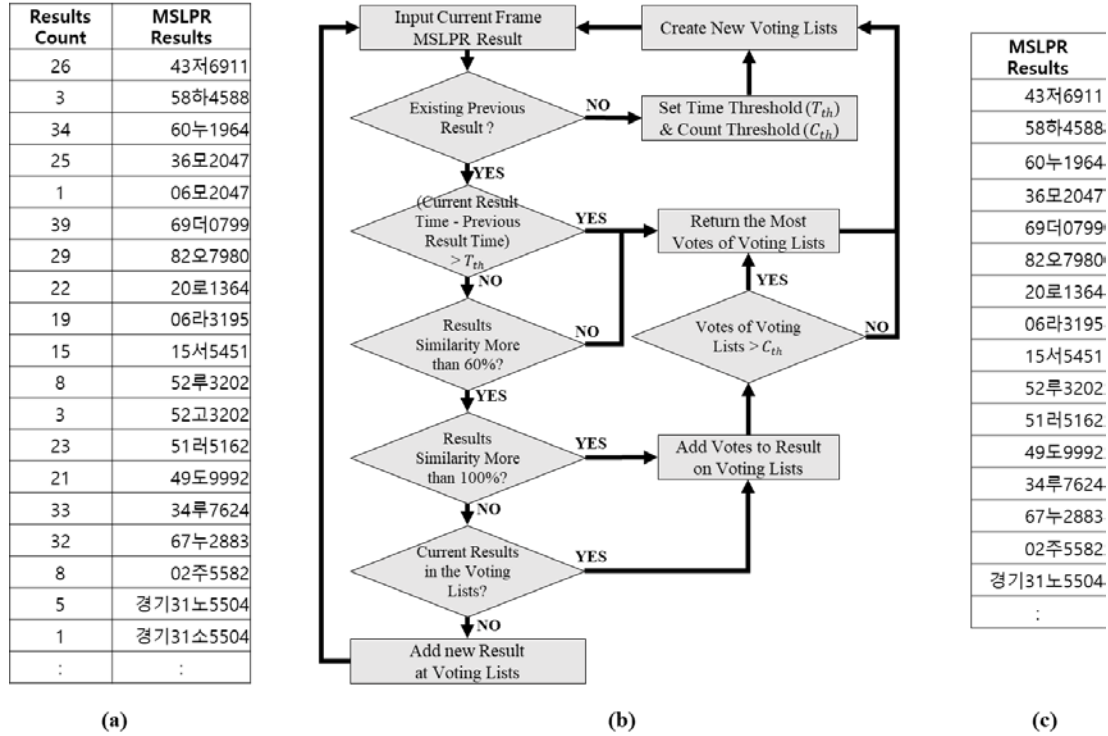


**Fig. 8.** A detailed procedure for the CRP, (a) the CRP results, (b) finding nearest feature data using KNN and (c) retrieving a label value from the training data

**Fig. 8** shows a detailed procedure for recognizing segmented characters (i.e., results from the CSP) using KNN. The CRP resizes width and height of the segmented character to a fixed size and removes margins. Then the CRP finds the best matching data using KNN from the trained data set. **Fig. 8** (b) shows the matches found according to different  $K$ . The CRP returns a label value paired with the best matching data and repeats this process with remaining segmented characters.

### 3.6 Result Validation

In the result validation process (RVP), the MSLPR system verifies intermediate results from the CRP against the identified LP structure (i.e., SLP or DLP) to confirm the result as the final. More specifically, the RVP examines whether the recognized result (i.e., LP numbers and characters in plain text) complies with all country-specific or use case specific style such as a number of alphanumeric characters, positions of characters and styles of characters. In the case of passenger cars in Korean, the LP format is composed of 6 digits and 1 Korean character. From left to right, the plate has the vehicle class, Korean character prefix, and four-digit serial number. The MSLPR system of Korean LP verifies CRP results using the format. When the MSLPRS uses video inputs, the system recognizes LPs from each frame to yield duplicate results. In this case, we use a voting method to return LP numbers and characters with most votes to mitigate false positive or noise data. **Fig. 9** shows a detailed procedure for voting method.



**Fig. 9.** A detailed procedure for voting method in RVP, (a) a part of duplicate CRP results by each frame, (b) a detailed procedure for the voting method and (c) final results with most votes

## 4. Multi-Style LP Experiments

In this section, we present results from two sets of experiments that cover six Korean LP styles and one style from Arizona State to show multi-style LP recognition using our proposed MSLPRS. In the first set of experiment, we used a 50-minute video that contains 138 vehicles moving in one direction to evaluate the MSLPRS for Korean LPs. We specified one ROI ( $500 \times 300$ ) in  $1920 \times 1080$  (i.e., Full HD) input images. In our experiment results, we show intermediate results and processing time for each LPCDP, LPSCP, CSP and CRP, respectively. For each frame of the input video, our MSLPRS recognizes LPs and confirms the result by the RVP. In the second set of experiments, we show extensibility of the proposed MSLPRS by showing LPR results (i.e., LP detection, character segmentation, character recognition processes, respectively) on one style of the U.S. LPs (i.e., from Arizona) and the newly established 2018 Korean LP specification.

### 4.1 Results of LP Candidates Detection

The LPCDP processes an input video frame-by-frame. The LPCDP finds LP candidates from each frame without detecting whether a vehicle is present in the frame. The LPCDP outputs LP candidates as shown in **Fig. 10** that may contain non-LP data. The MSLPRS removes detected non-LP data in later processes.



**Fig. 10.** Results of LPCDP that contain non-LP data

Consecutive and intermediate results in the LPCDP are depicted in **Fig. 11** and **Fig. 12**, respectively. **Fig. 11** shows results for the LPCDP on a style based on SLP with black characters. **Fig. 12** shows results for the LPCDP on a style based on DLP with non-black characters. On average, the LPCDP took 27 ms using our 50-minute video input.



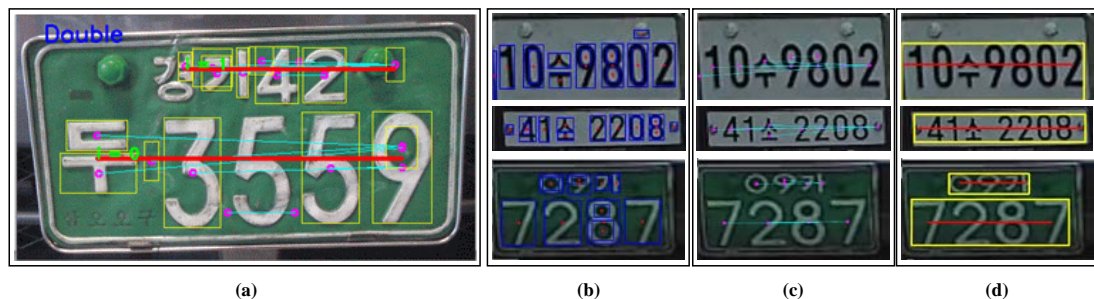
**Fig. 11.** Results of the LPCDP with type 1 (i.e., black characters) LPs



**Fig. 12.** Results of LPCDP with type 2 (i.e., non-black characters) LPs

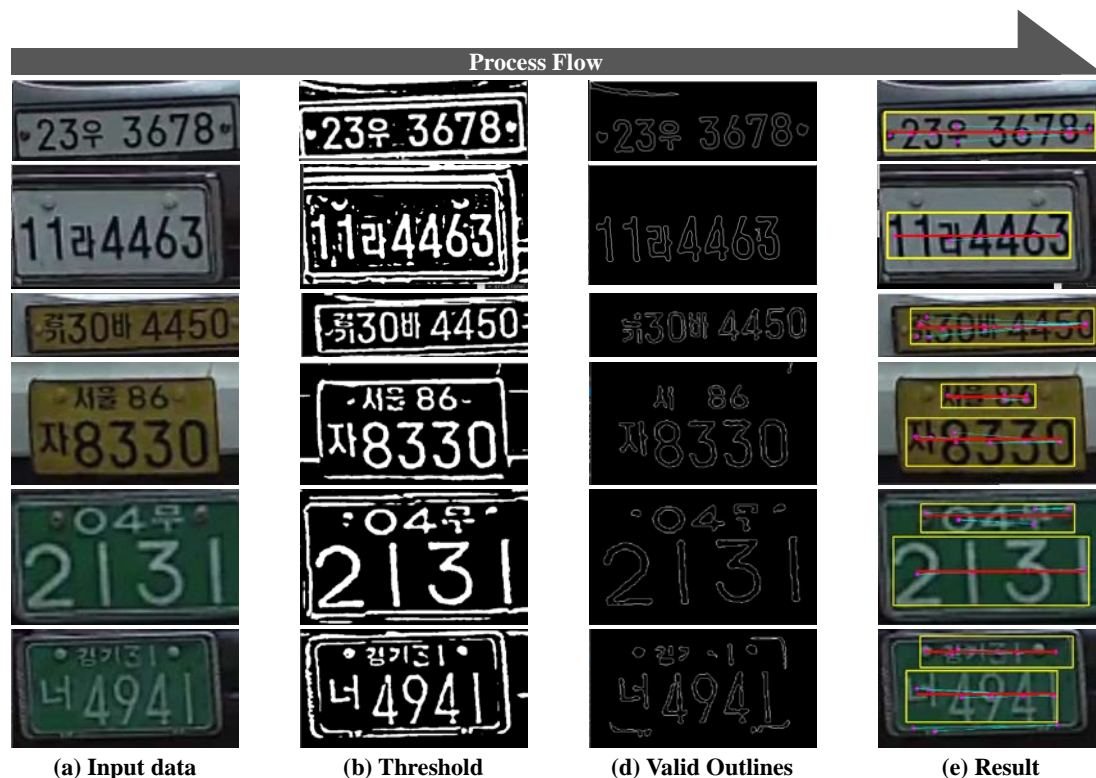
#### 4.2 Results of LP Structure Classification

**Fig. 13** shows intermediate steps and results for classifying an LP into SLP type or DLP type. The LPSCP finds valid data as shown in **Fig. 13** (b) and finds a point that forms an angle within a threshold with the horizontal axis. **Fig. 13** (d) shows results for the LPSCP on SLP and DLP structures. Magenta points in **Fig. 13** (a) represent centroid points within the threshold angle and cyan lines represent lines connecting those points. Red lines show a center-line for centroid points within the threshold angle. **Fig. 14** shows a detailed procedure for the LPSCP. The LPSCP took 10 ms on average.



**Fig. 13.** Classifying SLP and DLP structures in the LPSCP, (a) result of LPSCP, (b) centroid points of valid outlines, (c) pairing results within the threshold angle and (d) result of finding character areas using the LPSCP result

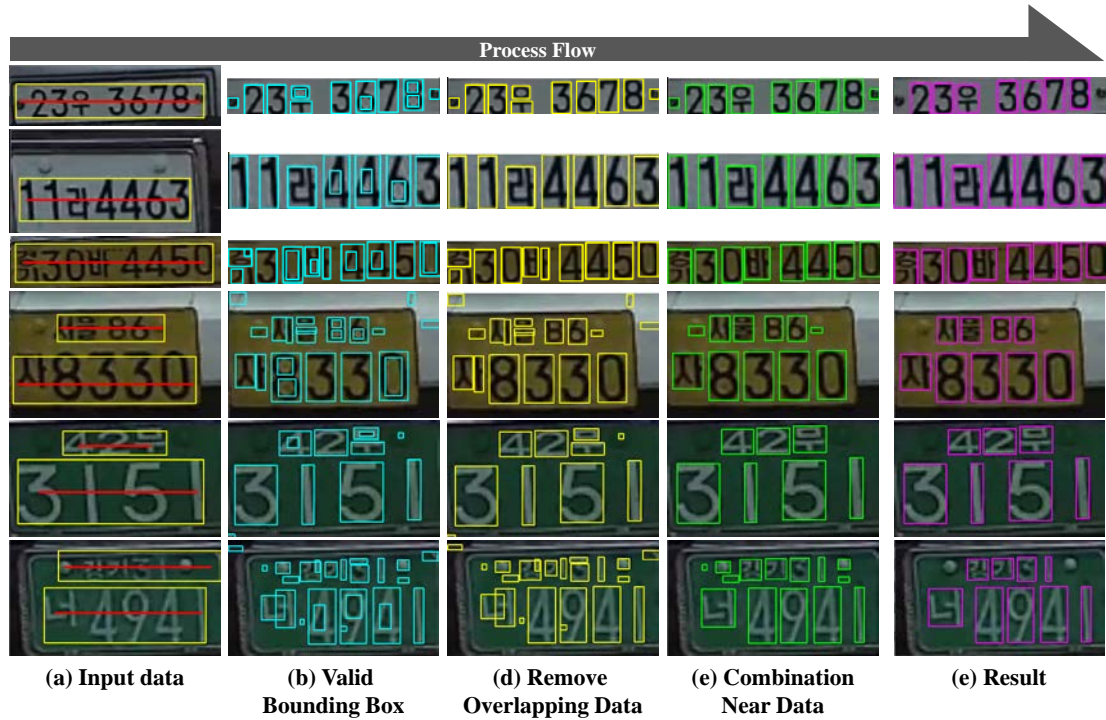




**Fig. 14.** Intermediate steps and results for the LPSCP, (a) input data to the LPSCP, (b) after applying the threshold, (c) valid outlines found and (d) LP structure classification results

### 4.3 Results of character segmentation and recognition

The CSP uses results from the LPSCP to calibrate parameters and to specify character areas. Since the LPCDP detects non-character areas as well, subsequent processes need to refine the detected areas for character recognition. When an LP's structure is identified as SLP, this structural information is exploited to find an area for character recognition. Similarly, when an LP's structure is identified as DLP, this structural information is exploited to separate two lines (i.e., upper and lower). To segment characters, the CSP uses different variables such as minimum and maximum values of contour area, box area, width/height ratio, width and height according to the identified LP structure. **Fig. 15** (a) shows a detailed procedure for using results from the LPSCP by finding bounding boxes for characters and segmenting upper and lower lines. Results in **Fig. 15** (d) are used in the CRP. To recognize successfully segmented character areas as shown in **Fig. 15** (e), we used more than 600 images of actual LP images for training. The CSP and the CRP altogether took 12 ms on average.



**Fig. 15.** A detailed procedure for the CSP, (a) inputs to the CSP, (b) valid bounding boxes are found, (c) overlapping areas are removed, (d) near bounding boxes are combined and (e) the final CSP results

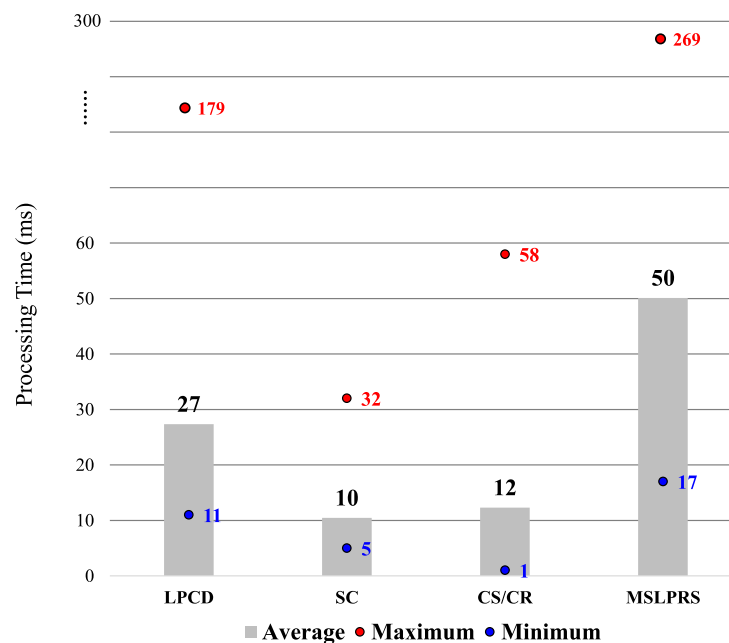
With our proposed MSLPRS, the frame-by-frame LPSCP may detect same LPs several times when a video is used as an input. We used a 50-minute video consisted of approximately 90,000 frames. Using this input data, the MSLPRS recognized 128,221 LP numbers correctly from 128,232 attempted LP numbers, yielding an accuracy of 99.99%. In addition, the MSLPRS correctly recognized 2,987 Hangul characters from 3,026 attempted characters yielding an accuracy of 98.71%. LP characters that represent regional area, we detected 17 cases and recognized them all correctly, yielding a 100% accuracy. Overall, our MSLPRS (more accurately, the CSP and the CSP) has 99.96% character recognition accuracy. We present a partial list of the CSP/CRP results on [Table 2](#).



**Table 2.** Numbers and Hangul character recognition accuracy of the CSP/CRP

Number	Total	Error	%	Hangul	Total	Error	%
0	11,527	3	99.97	가	107	0	100
1	11,925	2	99.98	라	219	8	96.34
2	14,002	0	100	더	147	0	100
3	13,737	1	99.99	러	150	0	100
4	12,750	2	99.98	어	104	0	100
5	12,512	0	100	저	152	1	99.34
6	12,770	2	99.99	로	166	0	100
7	13,671	0	100	구	140	0	100
8	12,584	2	99.99	무	127	2	99.11
9	12,754	0	100	수	198	0	100

When the frame-based LPCDP yields duplicate results, they are refined and compensated with the subsequent voting method. For example, we detected the same LP 95 times in our video input. In the voting method, we consider vehicles' spatiotemporal information (i.e., driving direction and moving time). The voting method return the most frequently detected LP result as the final result. We confirmed that all 138 vehicles and their LPs in our input video are successfully detected, segmented and recognized.





**Fig. 16.** A breakdown of processing time for the MSLPRS



**Fig. 16** shows a breakdown of processing time for the MSLPRS in LP detection, LP structure classification, character segmentation/recognition processes and overall MSLPRS performance. Since LPR systems are deployed and evaluated in various environments with different test data, we can not directly compare our MSLPRS performance with other LPR systems. Nevertheless, we report our MSLPRS performance and experiment setting in reference to previous work for indirect comparison along with **Table 1**. Our MSLPRS showed the overall LPR processing time under 50 ms and LP characters recognition rate of over 99% that match or sometimes outperform other LPR systems as shown in **Table 3**. **Table 4** shows 6 different LP styles recognized by the MSLPRS in the experiment.

**Table 3.** Multi-style LP detection

System (Year)	LP Target Country	Experiment Setting	System Performance
Yun et al. (2017) [11]	Korea	CPU: i5-5200U 2.2Ghz RAM: 8.00GB Visual Studio 2010, OpenCV 2.1.0, Matlab 2016b Input: 500 moving images	88.4 %
Tabrizi and Cavus (2016) [12]	Iran	No report Input: 257 images (1024 × 768)	CR: 97.03 %
Sarker et al. (2014) [13]	Korea	CPU: 3.10-GHz Intel Core i3-2100 RAM: 4GB Visual studio 2010 Input: 1800 images	LPD: 98.38 % Time : 54.64 ms
Deb et al. (2009) [14]	Korea	CPU: Pentium IV 2.4 MHz RAM: 1GB Matlab R2008a Input: 40 images (640 × 480)	82.5% Time : No Report
Proposed MSLPRS	Korea/ the U.S.	CPU: i7-4790K 4.00GHz RAM: 16GB Visual Studio 2013, OpenCV 3.3.0 Input: 50m Video (1920 × 1080)	Tested and Recognize Vehicle LPs: 138 LPD: 27 ms TC: 10 ms CS/CR: 12 ms

**Table 4.** 6 multi-style LPs in the experiment (WP: white plate, YP: yellow plate, GP: green plate, BC: black character, WC: white character)

LP Structure	Layout Format (R: Region, D: Digit, C: Character)	LP Size (mm) & Color	Example	Count in Video
SLP	DD C DDDD	335 × 155 WP + BC		7
	DD C DDDD	520 × 110 WP + BC		104
	R DD C DDDD	520 × 110 YP + BC		1
DLP	DD C DDDD	335 × 170 GP + WC		12

	R DD C DDDD	335 × 170 GP + WC		13
	R DD C DDDD	335 × 170 YP + BC		1

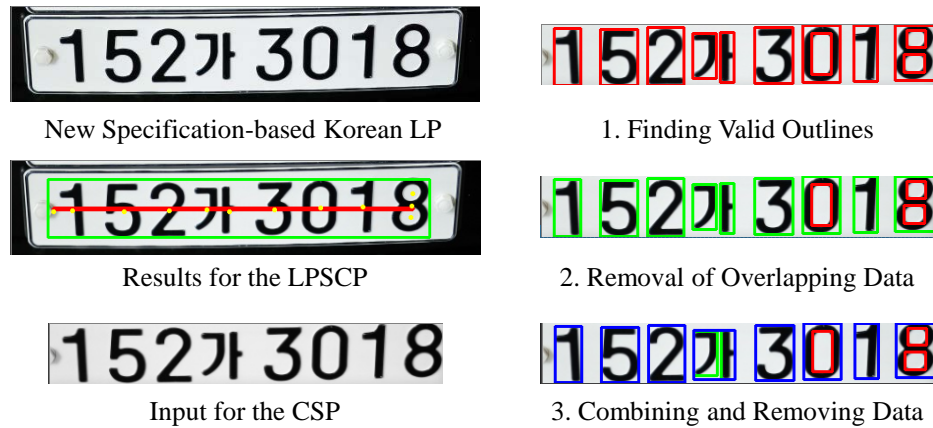
#### 4.4 Supporting Other Languages and Updated Specification

The proposed MSLPRS can be extended to recognize different LP styles from various countries and non-Characters data such as the national flags through training with additional LP data. We show extensibility of our system by recognizing LPs from Arizona State of the U.S. through additional learning. Fig. 17 shows the same MSLPR pipeline (i.e., LPCDP, LPSCP, CSP and CRP) used for recognizing Arizona LPs.



**Fig. 17.** Results of recognizing Arizona LPs using the MSLPR pipeline with additional training data, (a) input LP images, (b) results for the LPSCP, (c) results for the CSP and (d) results for the CRP

Also the proposed MSLPRS is readily usable with a recently published Korean LP specification in 2018. Due to the exhaustion of remaining unique LPs, Ministry of Land, Infrastructure and Transport of the Korean government announced a new LP specification that takes effect from September 2019. The new LP specification adds one more digit to secure two hundred million more LPs. To accompany this change, spacing of numbers and character size will be adjusted by the end of 2018. We ran a few images of LP that follows the new specification through our MSLPRS. Fig. 18 shows the same MSLPR pipeline (i.e., LPCDP, LPSCP, CSP and CRP) used for recognizing an LP with the new specification.



**Fig. 18.** Results of recognizing a new Korean specification-based LP

## 5. Conclusion

In this paper, we proposed and evaluated a MSLPRS to recognize multi-style LPs. Although LPR systems have been used in various fields in the last decade, different styles of LPs continue to cause challenges to LPR systems. To resolve these issues, we proposed a MSLPRS and its underlying pipeline for recognizing various LP styles of different colors and character properties. Our proposed MSLPRS trains and learns structural features of various styles of LPs. We demonstrated the performance our MSLPRS on Korean LPs as well as on Arizona LPs, showed extensibility of our system through additional learning and showed conformance to the new 2018 Korean LP specification. Through two sets of experiments, we showed that our MSLPRS successfully recognized all 138 vehicle LPs of 6 different styles from the 50-minute video, yielding over 99% accuracy on characters recognition all under 50 ms processing time, on average. Also the proposed LPR pipeline used in the MSLPRS, was readily usable through additional learning and extensible to an LP style with an added number. However, there are some limitations to our work. Our work is based on pre-calibrated and application-specific variables or parameters such as ROI size and LP width/height/ratio. Relaxing these constraints would allow the MSLPRS to be used in more diverse environment, but the MSLPR pipeline would take much longer time. Therefore, designing a strategy for balancing a trade-off between the performance and the flexibility of the MSLPRS and recognizing uncertain and contaminated LPs are two areas for further studies.

## References

- [1] S. Du, M. Ibrahim, M. Shehata and W. Badawy, "Automatic License Plate Recognition (ALPR): A State-of-the-Art Review," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 2, pp. 311-325, February, 2013. [Article \(CrossRef Link\)](#).
- [2] T. Naito, T. Tsukada, K. Yamada, K. Kozuka and S. Yamamoto, "Robust license-plate recognition method for passing vehicles under outside environment," *IEEE Transactions on Vehicular Technology*, vol. 49, no. 6, pp. 2309-2319, November, 2000. [Article \(CrossRef Link\)](#).
- [3] Sunghoon Kim, Daechul Kim, Younbok Ryu and Gyeonghwan Kim, "A robust license-plate extraction method under complex image conditions," in *Proc. of 16th Int. Conf. on Pattern Recognition*, pp. 216-219, August 11-15, 2002. [Article \(CrossRef Link\)](#).

- [4] Jun-Wei Hsieh, Shih-Hao Yu and Yung-Sheng Chen, "Morphology-based license plate detection from complex scenes," in *Proc. of 16th Int. Conf. on Pattern Recognition*, pp. 176-179, August 11-15, 2002. [Article \(CrossRef Link\)](#).
- [5] Ching-Tang Hsieh, Yu-Shan Juan and Kuo-Ming Hung, "Multiple license plate detection for complex background," in *Proc. of 19th Int. Conf. on Advanced Information Networking and Applications*, pp. 389-392, March 28-30, 2005. [Article \(CrossRef Link\)](#).
- [6] K. S. Raghunandan, P. Shivakumara, H. A. Jalab, R. W. Ibrahim, G. H. Kumar, U. Pal and T. Lu, "Riesz Fractional Based Model for Enhancing License Plate Detection and Recognition," *IEEE Transactions on Circuits and Systems for Video Technology, Early Access*, 2017. [Article \(CrossRef Link\)](#).
- [7] Shyang-Lih Chang, Li-Shien Chen, Yun-Chung Chung and Sei-Wan Chen, "Automatic license plate recognition," *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, no. 1, pp. 42-53, March, 2004. [Article \(CrossRef Link\)](#).
- [8] C. N. E. Anagnostopoulos, I. E. Anagnostopoulos, V. Loumos and E. Kayafas, "A License Plate-Recognition Algorithm for Intelligent Transportation System Applications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 3, pp. 377-392, September, 2006. [Article \(CrossRef Link\)](#).
- [9] Y. Wen, Y. Lu, J. Yan, Z. Zhou, K. M. von Deneen and P. Shi, "An Algorithm for License Plate Recognition Applied to Intelligent Transportation System," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 830-845, September, 2011. [Article \(CrossRef Link\)](#).
- [10] Byung-Gil Han, Jong Taek Lee, Kil-Taek Lim and Yunsu Chung, "Real-Time License Plate Detection in High-Resolution Videos Using Fastest Available Cascade Classifier and Core Patterns," *ETRI Journal*, vol. 37, no. 2, pp. 251-261, April, 2015. [Article \(CrossRef Link\)](#).
- [11] Jong-Ho Yun, Myung-Ryul Choi and Sang-Sun Lee, "An image enhancement method for extracting multi-license plate region," *KSII Transactions on Internet and Information Systems*, vol. 11, no. 6, pp. 3188-3207, June, 2017. [Article \(CrossRef Link\)](#).
- [12] Sahar S. Tabrizi and Nadire Cavus, "A Hybrid KNN-SVM Model for Iranian License Plate Recognition," *Procedia Computer Science*, vol. 102, pp. 588-594, 2016. [Article \(CrossRef Link\)](#).
- [13] Md. Mostafa Kamal Sarker, Sook Yoon and Dong Sun Park, "A Fast and Robust License Plate Detection Algorithm Based on Two-stage Cascade AdaBoost," *KSII Transactions on Internet and Information Systems*, vol. 8, no. 10, pp. 3490-3507, October, 2014. [Article \(CrossRef Link\)](#).
- [14] Kaushik Deb, Hyun-Uk Chae and Kang-Hyun Jo, "Vehicle License Plate Detection Method Based on Sliding Concentric Windows and Histogram," *Journal of Computers*, vol. 4, no. 8, pp. 771-777, August, 2009. [Article \(CrossRef Link\)](#).
- [15] Mei Yu and Yong Deak Kim, "An approach to Korean license plate recognition based on vertical edge matching," in *Proc. of 2000 IEEE Int. Conf. on Systems, Man, and Cybernetics*, pp. 2975-2980, October 8-11, 2000. [Article \(CrossRef Link\)](#).
- [16] Veronica Ong and Derwin Suhartono, "Using K-Nearest Neighbor in Optical Character Recognition," *ComTech*, vol. 7, no. 1, pp. 53-65, 2016. [Article \(CrossRef Link\)](#).
- [17] G. Kumar and P. K. Bhatia, "A Detailed Review of Feature Extraction in Image Processing Systems," in *Proc. of 4th Int. Conf. on Advanced Computing & Communication Technologies*, pp. 5-12, February 8-9, 2014. [Article \(CrossRef Link\)](#).
- [18] Muhammad 'Arif Mohamad, Haswadi Hassan, Dewi Nasien and Habibollah Haron, "A Review on Feature Extraction and Feature Selection for Handwritten Character Recognition," *International Journal of Advanced Computer Science and Applications*, vol. 6, no. 2, pp. 204-212, 2015. [Article \(CrossRef Link\)](#).
- [19] A. Coates, B. Carpenter, C. Case, S. Satheesh, B. Suresh, T. Wang, D. J. Wu and A. Y. Ng, "Text detection and character recognition in scene images with unsupervised feature learning," in *Proc. of Int. Conf. on Document Analysis and Recognition*, pp. 440-445, September 18-21, 2011. [Article \(CrossRef Link\)](#).



- [20] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu, “A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise,” in *Proc. of 2nd Int. Conf. on Knowledge Discovery and Data Mining*, pp. 226-231, August 2-4, 1996.
- [21] G. Shakhnarovich, T. Darrell and P. Indyk, “Nearest-Neighbor Methods in Learning and Vision: Theory and Practice,” *The MIT Press*, 2006.
- [22] B. Gérard and L. Devroye, “Lectures on the Nearest Neighbor Method,” 2015.



**Soungsill Park** is a researcher at Korea Electronics Technology Institute (KETI), Seoul, Korea. She received her M.S. in Image Science and Arts (Image Engineering - computer graphics/virtual reality) from Chung-ang University in 2016. She received her B.S. in Engineering (department of digital media) from Duksung Women's University in 2014. Her current research interests are in the areas of 2D/3D image processing, pattern recognition and machine learning.



**Hyoseok Yoon** is a senior researcher at Korea Electronics Technology Institute (KETI), Seoul, Korea. He received his Ph.D. and M.S. degrees in Information and Communications (Computer Science and Engineering) from Gwangju Institute of Science and Technology (GIST), in 2012 and 2007, respectively. He received his B.S. in Computer Science from Soongsil University in 2005. Before joining KETI in 2014, he was a research associate at KAIST Culture Technology Research Institute and was a researcher at GIST Culture Technology Institute from 2012 to 2013. He was an intern researcher in Microsoft Research Asia in 2009. His research interests include ubiquitous computing (context-awareness, wearable computing) and HCI (mobile and wearable UI/UX, MR/AR/VR interaction and contents).



**Seho Park** is a managerial researcher at Korea Electronics Technology Institute (KETI), Seoul, Korea. Before joining KETI in 2005, he was with I&C Technology and Samsung Electronics, where he worked in projects relevant to digital broadcasting and wireless networks. He received his B.S. and M.S. degrees in Electrical Engineering from Kyungpook National University in 1998 and 2000, respectively. His current research interests are in the areas of 2D/3D image processing, pattern recognition and AR/MR Interaction technology.