



Article

# License Plate Image Generation using Generative Adversarial Networks for End-To-End License Plate Character Recognition from a Small Set of Real Images

Byung-Gil Han <sup>1</sup>, Jong Taek Lee <sup>1</sup>, Kil-Taek Lim <sup>1</sup> and Doo-Hyun Choi <sup>2</sup>,\*

- Electronics and Telecommunications Research Institute, Daegu 42994, Korea
- <sup>2</sup> School of Electronics Engineering, Kyungpook National University, Daegu 41566, Korea
- \* Correspondence: dhc@ee.knu.ac.kr

Received: 2 March 2020; Accepted: 11 April 2020; Published: 16 April 2020



Abstract: License Plate Character Recognition (LPCR) is a technology for reading vehicle registration plates using optical character recognition from images and videos, and it has a long history due to its usefulness. While LPCR has been significantly improved with the advance of deep learning, training deep networks for LPCR module requires a large number of license plate (LP) images and their annotations. Unlike other public datasets of vehicle information, each LP has a unique combination of characters and numbers depending on the country or the region. Therefore, collecting a sufficient number of LP images is extremely difficult for normal research. In this paper, we propose LP-GAN, an LP image generation method, by applying an ensemble of generative adversarial networks (GAN), and we also propose a modified lightweight YOLOv2 model for an efficient end-to-end LPCR module. With only 159 real LP images available online, thousands of synthetic LP images were generated by using LP-GAN. The generated images not only looked similar to real ones, but they were also shown to be effective for training the LPCR module. As a result of performance tests with 22,117 real LP images, the LPCR module trained with only the generated synthetic dataset achieved 98.72% overall accuracy, which is comparable to that of training with a real LP image dataset. In addition, we improved the processing speed of LPCR about 1.7 times faster than that of the original YOLOv2 model by using the proposed lightweight model.

**Keywords:** license plate image generation; ensemble data; segmentation-free; end-to-end recognition; GAN; ALPR

# 1. Introduction

License plate (LP) character information is uniquely assigned so that each vehicle on the road can be identified. Therefore, it is widely used for vehicle recognition in situations such as toll charges on highways, speed and signal violations, and illegal parking detection [1–3]. Due to the serious number of traffic-related problems caused by the rapid increase in vehicles, research is currently underway to improve the traffic environment and LP information is used as an important source of information [4]. For this reason, the study of Automatic LP Recognition (ALPR) has been underway for a long time and is continuing to this day. License plate character recognition (LPCR) is a technology for reading vehicle registration plate character information using optical character recognition. In the conventional LPCR process, each character is segmented from the LP image and the character recognition is performed for individual characters. The end-to-end LPCR method using an object detector based on convolutional neural network (CNN) can recognize LP character information by performing character segmentation and character recognition simultaneously from the LP image. However, this method requires a significantly large number of LP images and their annotations of character information for network

Appl. Sci. 2020, 10, 2780 2 of 16

training. Unlike other public datasets of vehicle information, each LP has a unique combination of characters and numbers depending on the country or the region.

In the following situation, as an example, it is necessary to develop an LPCR module of the ALPR system for Korean LPs. After determining a method and an algorithm for developing the LPCR, a sufficient sample of real LP images is needed for development and testing. If machine learning or deep learning in the algorithm is decided to use for the LPCR, more LP image data are needed, but unfortunately, it is extremely difficult to obtain enough real LP image data. Since it takes a lot of time, effort, and money to obtain enough LP image data to develop a practical LPCR module, we can instead search for public LP datasets. Table 1 shows the available image datasets related to LP from Caltech Cars dataset [5] released from 1999 to the present. Most of the datasets have hundreds of images and UFPR-ALPR [6] dataset offers 4500 images. However, these LP images are not suitable for training LPCR of Korean LPs that consist of numbers and Korean characters, because these datasets are composed of numbers and English or Chinese characters. Thus, Korean LP images had to be collected through web-scraping, but only a few hundred were available.

Dataset	Number of Images	Country	Year
Caltech Cars [5]	126	USA	1999
EnglishLP [7]	509	Europe	2003
UCSD-Stills [8]	291	USĀ	2005
ChineseLP [9]	411	China	2012
AOLP [10]	2049	Taiwan	2013
OpenALPR-EU [11]	108	Europe	2016
SSIG-SegPlate [12]	2000	Brazil	2016
UFPR-ALPR [6]	4500	Brazil	2018

Table 1. Public license plate image datasets.

To solve this problem concerning insufficient data, we propose the generation of LP images using a Generative Adversarial Network (GAN)-based image-to-image translation (LP-GAN) model. The advantages of generating LP images using LP-GAN are (1) they look similar to real LP images and (2) they can be used as training data to improve the performance of the LPCR in the ALPR system.

The main contribution is three-fold. First, an LP generator based on GAN was made by a small set of real LP images, and generating realistic LP images that have the desired character information for use as training data for the LPCR module of the ALPR system. LP-GAN can be trained using an extremely limited amount of LP image data and can then generate realistic LP images. The generated LP images were used as training data for the LPCR, and were confirmed through experiments that character information could be effectively recognized from a real LP image. In addition, it is shown that the performance of the LPCR could be improved by an ensemble of LP images generated by various LP-GAN generators rather than a single one. Figure 1 shows sample LP images generated by various GAN methods and real LP images. LP-GAN can easily generate LP images of any character combinations, and the generated images look very real.

Second, an object detector based on the Convolutional Neural Networks (CNN) was developed which is able to perform character segmentation and character recognition simultaneously. These tasks are carried out separately in the traditional ALPR systems. For LPCR methods using conventional optical character recognition approaches, the segmentation of each character in the LP image must precede the character recognition. Therefore, successful character segmentation greatly affects the LPCR performance. However, the CNN-based object detector of an LPCR is performed with a segmentation-free end-to-end manner that is not affected by character segmentation problems.

Third, an extensive test of algorithms was performed with 22,117 real LP images under the various conditions. The real LP images were not used in any of the training phases of LP-GAN-based LPCR module. As an LPCR module trained with the generated dataset using LP-GAN achieved comparable

Appl. Sci. 2020, 10, 2780 3 of 16

accuracy to that trained with a real LP image dataset, it was successfully shown the high feasibility of data generation using GAN for LP images.

The rest of this paper is organized as follows. Section 2 reviews the GAN models studied so far, especially image-to-image translation methods and recent studies related to LP recognition technology, including traditional LP recognition methods. The three GAN-based image-to-image translation methods used in this study and the generation of Korean LP images using LP-GAN are discribed in Section 3. The CNN-based object detector that can perform character segmentation and character recognition simultaneously in a segmentation-free end-to-end manner is discussed in Section 4. Section 5 describes the dataset configured for the experiments, give details on the procedure of generating Korean LP images using the three LP-GANs and using the generated LP images as the training data for the LPCR module, and finally report the performance of the LPCR on the real LP images. Conclusions on this study are given in Section 6.

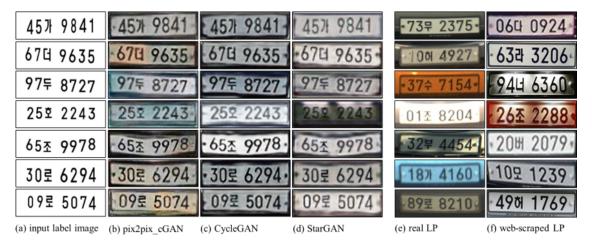


Figure 1. Sample LP images by various LP-GAN generators and real LP images.

# 2. Related Works

#### 2.1. Image-to-Image Translation

Goodfellow et al. [13] proposed GAN models in which the generators and discriminators that are adversarial toward each other gradually improve each other's performance so that the generators learn to generate data that is as close as possible to the final target data. Mirza et al. [14] suggested an improvement on GAN called Conditional GAN (cGAN). Since GANs generate the output data from random noise input data, control over the output is difficult. However, cGAN can control the output partly by adding conditions to the GAN. Since then, cGAN has been applied in many fields such as image generation [15–18], image domain transfer [19–22], image super-resolution [23,24], and image editing [25].

For image-to-image translation problems, Isola et al. [19] introduced cGAN to solve the problem of blurring of the resultant image as a result of pixel-to-pixel translation based on CNN. Zhu et al. [20] proposed an unpaired image-to-image translation model to solve the problem that existing image-to-image translation models need the paired images of input and output as training data. Existing image-to-image translation models can successfully translate images between two domains but scalability and robustness are limited for more than two domains; Choi et al. [21] suggested StarGAN to solve the problem of translating images between multiple domains in a single model.

In this study, LP images were generated using the three state-of-the-art GAN-based image-to-image translation models mentioned earlier. Moreover, it was verified through experiments that the generated LP images are similar to real ones and that they were sufficient for use as training data for the LPCR.

Appl. Sci. 2020, 10, 2780 4 of 16

#### 2.2. Automatic License Plate Recognition

Traditional ALPR systems consist of three steps: LP detection, LP character segmentation, and LPCR. Methods for LP detecting are based on using boundary and edge information [26,27], global image information [28], texture features [29], color features [30], character features [31], and core patterns [32]. LP character segmentation methods include pixel connectivity [33], projection profiles [34], prior knowledge of characters [35], character contours [36], and multiple-method binarization schemes [37]. Finally, LPCR methods include using pixel data directly [38], extracted features [39], and neural networks (NNs) [40].

These traditional ALPR systems are segmentation-based algorithms that are heavily influenced by the performance of each stage due to various environmental factors such as distortion, contamination, illumination, and noise. In recent studies, segmentation-free algorithms based on Deep NNs (DNNs) were proposed to overcome these environmental factors [41–43].

#### 3. License Plate Image Generation Via LP-GAN

This section discusses the generation of LP images using the three state-of-the-art GAN-based image-to-image translation methods.

#### 3.1. GAN Approaches

Existing CNN-based image-to-image translation methods have a problem in that the resultant image is not photo-realistic because the loss function uses the average of each pixel loss as the total loss. To solve this problem, Isola et al. [19] proposed an image-to-image translation algorithm via pix2pix\_cGAN that uses U-Net [44] as a generator network to reduce the loss of information in an encoder-decoder structure. PatchGAN [45] was used in the discriminator network to improve the detail of the resultant image by using loss-per-patch. Pix2pix\_cGAN is trained with paired datasets, although in reality, unpaired data is used more often than paired data. Zhu et al. [20] suggested CycleGAN with cyclic consistency to enable the learning of unpaired datasets and to enable image-to-image translation. This approach uses Resnet [46] as a generator network, LSGAN [47] for the loss, and PatchGAN, the same discriminator network used in pix2pix\_cGAN.

The existing image-to-image translation methods require k(k-1) generator networks to perform image translation between k multi-domains. To improve this, Choi et al. [21] implemented StarGAN, which enables image-to-image translation between multi-domains with a single generator. Domain classification loss and a reconstruction loss are used in this method for multi-domain image-to-image translation. A target domain label consisting of a binary or one-hot vector is used to specify the target domain to which the input image is translated.

# 3.2. License Plate Image Generation

Currently, LP characters in Korea are composed of seven black characters on a white background, as shown in Figure 2a. The first two digits are the car-code for the type of vehicle, and the third Korean character is the use-code specifying the use of the vehicle. The last four digits are the serial number. Figure 2b shows the character classes present in Korean LP. Numbers are defined as 10 classes from 0 to 9, and Korean characters are defined as 35 classes. In this study, an ID of C1 to C35 was assigned to each of the 35 Korean characters for convenience of expression.

As mentioned in the introduction, consider a situation that there is a small set of real LP images. By searching for 'license plate' through websites such as Google and SNS, we were able to scrape over 5,000 related images, out of which only 159 LP images were actually available. In the collected 159 LP image dataset (Web\_159), there are 954 numbers and 159 Korean characters. Figure 3a shows the character class distribution of the Web\_159 dataset. As shown in Figure 3a, there the C15 Korean character class is not present in the Web\_159 dataset, there was only one incidence each of classes C12, C18, C33, and C34.

Appl. Sci. 2020, 10, 2780 5 of 16

# 52 가 3108 (a) Example of Korean LP Form 0123456789 가거고구나너노누다더 도두라러로루마머모무

조주하허호

(b) Character Classes in Korean LP

Figure 2. Korean LP form and character classes in Korean LP.

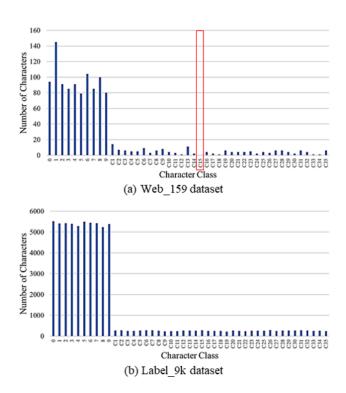


Figure 3. Character class distribution of Web\_159 and Label\_9k datasets.

To train LP-GAN for generating LP images, paired images were prepared: the target images (i.e., Web\_159) and the label images. As shown in Figure 4, the widths of the paired images were resized to 256 pixels and then zero-padded at the top and bottom to normalize them to  $256 \times 256$  pixels. The character string of the label image is the same character in the same position in the target image. Finally, the normalized paired images were input into the proposed LP-GAN as the training data.

For generating the LP images using the LP-GAN generator after training had been completed, 9000 input label images were made in which each LP character had a uniformly random distribution to create the Label\_9k dataset. Figure 3b shows the character class distribution in the Label\_9k dataset,

Appl. Sci. 2020, 10, 2780 6 of 16

in which all of them are distributed uniformly, including the C15 character class which is non-existent in the Web\_159 dataset. Figure 5 shows the process of generating the 9000 LP by 9000 input label image pairs using LP-GAN. After the input label images of the Label\_9k dataset had been converted to zero-padded normalized images, the latter were input into the LP-GAN generator, after which the output images were de-normalized to generate the final LP images.

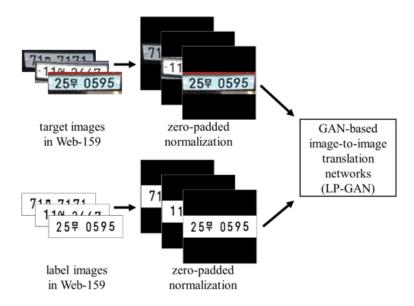


Figure 4. Training of GAN-based image-to-image translation networks.

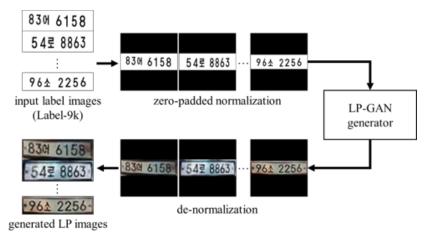


Figure 5. Generation of LP images using LP-GAN.

Figure 1 shows some examples of the LP image generated by the three state-of-the-art GAN-based LP-GAN generators. In row 6 and 7 of Figure 1, the C15 character as well translated as the other characters that exist in the Web\_159 dataset despite it not existing in the training dataset.

## 4. Segmentation-Free End-to-End LPCR By Object Detector

This section discusses how the end-to-end object detector can be used as a segmentation-free end-to-end LPCR method. Redmon et al. [48] proposed a novel state-of-the-art real-time object detector (YOLOv2) that can detect 9000 different categories. While existing CNN-based object detection models such as Faster R-CNN [49] perform region proposals first and then classify each set of boundary boxes, the YOLOv2 detector considers region proposals and class probabilities as one regression problem and simultaneously performs the location and classification of objects in a single CNN. Using this idea

Appl. Sci. 2020, 10, 2780 7 of 16

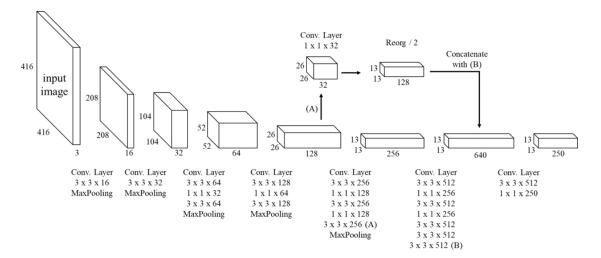
of the YOLOv2 detector, it only needs to detect 45 character classes in the LP images. Accordingly, the whole LPCR process is carried out at once.

In practical ALPR systems, it is necessary to minimize the processing time and the size of required GPU memory in the LPCR stage, because the system must carry out other important processes such as acquiring images, storing recognition results, and communicating with a remote server. In addition, when considering the cost, a system with a powerful but expensive GPU processor may not always be available to run on the ALPR system, so the architecture needs to be as cost-effective and lightweight as possible. To improve these problems, we propose a modified YOLOv2 model with half of the CNN-layer filters in the YOLOv2 detector architecture proposed by Redmon et al. [48] as the LPCR method. The architecture and structure of the proposed LPCR module (i.e., modified YOLOv2) is given in Table 2 and Figure 6, respectively. The modified YOLOv2 model outputs a  $13 \times 13 \times 512$ feature map from a 416  $\times$  416 pixel 3-channel image through five steps of convolutional and maxpool layers (Figure 6B). Next, the  $26 \times 26 \times 256$  feature map layers pass through and are reshaped into  $13 \times 13 \times 128$  feature map layers, as shown in Figure 6A, and become  $13 \times 13 \times 640$  feature map layers by concatenation with the output from Figure 6B. In the last step,  $13 \times 13 \times 250$  layers are output for the location and classification of the 45 character classes that is needed to detect (i.e., the prediction of 5 boxes with 5 coordinates each and 45 classes per box =  $5 \times (5 + 45)$ ). If sufficient amount of training data were provided, the proposed YOLOv2 detector as a segmentation-free end-to-end LPCR method could achieve high performance for detecting 10 number classes and 35 character classes, which was experimentally confirmed.

Table 2. Modified YOLOv2 model architecture.

No.	Layer Type	Filters	Size / Stride	Output
0	Convolutional	16	3 × 3	$416 \times 416 \times 16$
1	Maxpool	-	$2 \times 2 / 2$	$208 \times 208 \times 16$
2	Convolutional	32	$3 \times 3$	$208 \times 208 \times 32$
3	Maxpool	-	$2 \times 2 / 2$	$104 \times 104 \times 32$
4	Convolutional	64	$3 \times 3$	$104 \times 104 \times 64$
5	Convolutional	32	$1 \times 1$	$104 \times 104 \times 32$
6	Convolutional	64	$3 \times 3$	$104 \times 104 \times 64$
7	Maxpool	-	$2 \times 2 / 2$	$52 \times 52 \times 64$
8	Convolutional	128	$3 \times 3$	$52 \times 52 \times 128$
9	Convolutional	64	$1 \times 1$	$52 \times 52 \times 64$
10	Convolutional	128	$3 \times 3$	$52 \times 52 \times 128$
11	Maxpool	-	$2 \times 2 / 2$	$26 \times 26 \times 128$
12	Convolutional	256	$3 \times 3$	$26 \times 26 \times 256$
13	Convolutional	128	$1 \times 1$	$26 \times 26 \times 128$
14	Convolutional	256	$3 \times 3$	$26 \times 26 \times 256$
15	Convolutional	128	$1 \times 1$	$26 \times 26 \times 128$
16	Convolutional	256	$3 \times 3$	$26 \times 26 \times 256$
17	Maxpool	-	$2 \times 2 / 2$	$13 \times 13 \times 256$
18	Convolutional	512	$3 \times 3$	$13 \times 13 \times 512$
19	Convolutional	256	$1 \times 1$	$13 \times 13 \times 256$
20	Convolutional	512	$3 \times 3$	$13 \times 13 \times 512$
21	Convolutional	256	$1 \times 1$	$13 \times 13 \times 256$
22	Convolutional	512	$3 \times 3$	$13 \times 13 \times 512$
23	Convolutional	512	$3 \times 3$	$13 \times 13 \times 512$
24	Convolutional	512	$3 \times 3$	$13 \times 13 \times 512$
25	Route 16	-	-	$26 \times 26 \times 256$
26	Convolutional	32	$1 \times 1$	$26 \times 26 \times 32$
27	Reorg.	-	/ 2	$13 \times 13 \times 128$
28	Route 27 24	-	-	$13 \times 13 \times 640$
29	Convolutional	512	$3 \times 3$	$13 \times 13 \times 512$
30	Convolutional	250	$1 \times 1$	$13\times13\times250$

Appl. Sci. 2020, 10, 2780 8 of 16



**Figure 6.** Structure of the proposed YOLOv2 model.

#### 5. Experimental Section

In this section, the dataset configuration for the experiments is reported and the experimental results are discussed after describing the implementation details. At the experiments, quantitatively evaluation is performed in order to prove the usefulness of the LP images generated by proposed LP-GAN generators as training data for the LPCR. The experimental results showed that the LPCR trained with the LP images from the three LP-GAN generators outperformed the LPCR trained with the LP images from a single LP-GAN generator.

#### 5.1. Datasets

#### 5.1.1. Web-Scraped Real Images

As mentioned in Section 3.2, 159 real LP images were collected through web-scraping. The Web\_159 dataset was used to train the three LP-GAN generators. To compare the LPCR performance when using a small set of data, the Web\_159 dataset was also used to train the LPCR module.

## 5.1.2. Generated Datasets by LP-GAN

To test whether the LP images generated by the LP-GAN generators can be used as training data for the LPCR module of the ALPR system, we prepared several training datasets with various conditions. At first, after the Label\_9k dataset was input into the LP-GAN generator trained by pix2pix\_cGAN, CycleGAN, or StarGAN, a training LP dataset was obtained for each of them (pix2pix\_cGAN\_9k, CycleGAN\_9k, and StarGAN\_9k, respectively). Next, to test the performance of the LPCR according to the number of training data items, another three training datasets were prepared by randomly selecting 3,000 images from each of the pix2pix\_cGAN\_9k, CycleGAN\_9k, and StarGAN\_9k datasets (pix2pix\_cGAN\_3k, CycleGAN\_3k, and StarGAN\_3k, respectively). Last, to confirm whether the LP images from the three LP-GAN generators together enhanced the performance of the LPCR more than each on its own, two ensemble datasets were prepared from all three sources: the Ensemble\_9k dataset was combined with pix2pix\_cGAN\_3k, CycleGAN\_3k, and StarGAN\_3k and the Ensemble\_3k dataset was combined with 1000 images randomly selected from each of the pix2pix\_cGAN\_3k, CycleGAN\_3k, and StarGAN\_3k datasets.

# 5.1.3. Real Datasets for Comparison and Testing

Twenty-two thousand, one-hundred and seventeen real LP images captured by CCTV at more than 10 different locations were obtained. All of the LP images were labeled with the character information. 9000 of the LP images were randomly selected as the training dataset (Real\_9k), and the

Appl. Sci. 2020, 10, 2780 9 of 16

remaining 13,117 LP images comprised the test dataset (Test\_13k). In addition, for the same comparison, 3000 LP images were randomly selected from the Real\_9k dataset to configure the Real\_3k dataset and then likewise, 159 LP images were randomly selected from the Real\_3k dataset as the Real\_159 dataset for comparison with the Web\_159 dataset. The 13 datasets prepared for the experiments are summarized in Table 3.

Dataset	Description
Web_159	Real LP images from Web-Scraping
pix2pix_cGAN_9k CycleGAN_9k StarGAN_9k	9000 generated LP images by three state-of-the-art GAN based LP-GAN from Label_9k
pix2pix_cGAN_3k CycleGAN_3k StarGAN_3k	Randomly selected from each of pix2pix_cGAN_9k, CycleGAN_9k, StarGAN_9k
Ensemble_9k	Combined generated LP images. (pix2pix_cGAN_3k + CycleGAN_3k + StarGAN_3k)
Ensemble_3k	Combined randomly selected 1000 LP images from each of pix2pix_cGAN_3k, CycleGAN_3k, StarGAN_3k
Real_9k	9000 real LP images for training data
Real_3k	3000 real LP images randomly selected from Real_9k
Real_159	159 real LP images randomly selected from Real_3k
Test_13k	13,117 real LP images for LPCR testing

Table 3. 13 datasets for experiments.

#### 5.2. Implementation Details

This subsection presents the detailed setup of the system, algorithms, and frameworks used in the experiments. All experiments were performed on PC systems with Intel-i7 CPUs and NVIDIA Titan Xp GPUs. For the three state-of-the-art GAN-based image-to-image translation models (pix2pix\_cGAN, CycleGAN, and StarGAN) for generating LP images, the code published by each author was used [50,51]. The LPCR performance experiments were performed using the Darknet framework [48] and using the modified YOLOv2 model in which the number of CNN-layer filters is reduced by half.

#### 5.2.1. LP Generation

The code for pix2pix\_cGAN and CycleGAN is written in PyTorch [52] by the same group of researchers. Thus, the network model architecture is different but the training options are nearly identical. The size of the input images was set as 256 pixels, and since the training data was zero-padded and normalized to  $256 \times 256$  pixels, the preprocess option in the code was set to 'none'. Other options (number of iterations, batch size, learning rate, etc.) were set to the default values provided by the author. StarGAN, also written in PyTorch, is capable of multi-domain translation, but in this experiment, only used the translation between two domains, label images, and LP images. As was carried out for the other two models, the size of the input image was set as 256 pixels and the domain dimension was set as 2. Other parameters of StarGAN, such as the number of iterations, batch size, and learning rate, etc. were set to the default values provided by the author.

As mentioned previously, the pix2pix\_cGAN\_9k, CycleGAN\_9k, and StarGAN\_9k datasets were generated from the Label\_9k dataset using the three LP-GAN generators. Figure 1 shows some of the resultant images of each proposed LP-GAN generator for the same input label image. Figure 1a shows the input label images; Figure 1b–d, reveal the resultant images from each LP-GAN generator; and Figure 1e,f present the real LP images acquired by CCTV and web-scraping, respectively. From the results, it can be seen that the proposed LP-GAN generators could translate in the same style as the real LP while maintaining the character information of the input label images. In particular, even when the input label images contained character information that did not exist in the training data (such as

the C15 character class) were input into the generators, the character class was correctly translated exactly the same as the other existing character classes.

# 5.2.2. LP Recognition

The LPCR modules were trained by using the LP images generated through the proposed LP-GAN generators. Subsequently, the results of the experiments on recognizing the characters of the real LP images show that the LP images generated by the LP-GAN generators were similar to the real LP images and could be used as training data for the LPCR module in the ALPR system. The LPCR modules based on the modified YOLOv2 were trained with an input image size of  $416 \times 416$  pixels, a starting learning rate of 0.00025, a batch size of 64, weight decay of 0.0005, and momentum of 0.9. The training was conducted using four NVIDIA Titan Xp GPUs.

#### 5.3. Experimental Results

The 12 LPCR modules were trained using 12 datasets among the 13 datasets configured previously (except for Test\_13k). The accuracy of the trained LPCR modules was compared using the Test\_13k dataset.

The LPCR performance of each of the seven characters in a Korean LP was compared and then overall performance from complete LPs was compared. Since the character information of the LP is unique, each item of character information in the LP should be correctly recognized. Therefore, the performance of the LPCR module was evaluated based on the accurate recognition of all seven characters rather than the recognition performance of individual characters. The modified YOLOv2 detector proposed in this paper simultaneously performs the location and classification of the object of interest (i.e., the characters in the LP), but the LPCR module obtains the same result if the character classification is correct, even if the location of the character is incorrect. Hence, the experiments in this study did not consider the locational accuracy of the characters.

Table 4 reports the performance comparison results of the LPCR modules trained with the 12 different training datasets. All characters except for the third Korean character are number classes, and the recognition performance of numbers was over 99% with almost all of the comparison datasets. Since there are only 10 classes for numbers and the distinction between the types is obvious, the performance was high regardless of the dataset. In the case of the third Korean character, there are 35 classes, which makes identifying them more complex than for numbers, so there was a difference in performance per dataset. Therefore, the overall performance of LPCR strongly depends on the recognition performance for the Korean characters.

The accuracies of the Real\_9k, Real\_3k, and Real\_159 datasets with the proposed modified YOLOv2 detector were 99.78%, 99.72%, and 97.85%, respectively, reflecting high LPCR performance because the LPCR modules had been trained with real LP images and so more easily recognized. This means that the proposed modified YOLOv2 detector gave the LPCR sufficient recognition performance for use in the ALPR system. Provided that there are enough LP images to train the LPCR module, a high-performance LPCR module for the ALPR system could be developed with more than 99.7% overall accuracy.

The comparison results of the LPCR performance of the six LPCR modules trained with the six LP datasets generated by the three proposed LP-GAN generators are shown in Figure 7. As can be seen, the performance of the LPCR module trained with the LP images generated by the pix2pix\_cGAN-based LP-GAN was higher than the other two LPCR modules trained with CycleGAN or StarGAN. This means that among the three GAN-based image-to-image translation methods, pix2pix\_cGAN generated LP images that were more realistic than those from the other two methods. For the LP images generated via a single LP-GAN generator, the more items in the training data, the higher the recognition performance.

Appl. Sci. 2020, 10, 2780 11 of 16

Training Dataset	1st (num)	2nd (num)	3rd (char)	4th (num)	5th (num)	6th (num)	7th (num)	Overall
Web_159	99.85	99.87	94.69	99.89	99.92	99.86	99.86	94.45
Real_9k	99.97	99.98	99.84	99.98	99.98	99.99	99.98	99.78
Real_3k	99.96	99.95	99.80	99.98	99.98	99.99	99.98	99.72
Real_159	99.94	99.95	97.95	99.95	99.95	99.96	99.95	97.85
pix2pix_cGAN_9k	99.85	99.86	96.57	99.92	99.94	99.95	99.83	96.33
CycleGAN_9k	99.35	99.38	94.97	99.38	99.56	99.45	98.98	93.59
StarGAN_9k	99.43	99.36	95.21	99.48	99.45	99.61	99.45	94.23

99.89

99.29

99.48

99.93

99.90

99.91

99.46

99.56

99.95

99.94

99.92

99.51

99.41

99.95

99.92

99.80

99.13

99.17

99.88

99.81

93.91

89.48

92.13

98.72

95.56

94.16

90.56

93.25

99.02

95.94

pix2pix\_cGAN\_3k

CycleGAN\_3k

StarGAN\_3k

Ensemble 9k

Ensemble\_3k

99.87

99.13

99.51

99.86

99.80

99.86

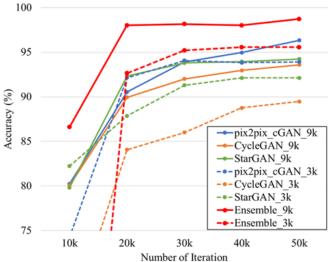
99.32

99.29

99.92

99.86

Table 4. Performance comparison of LPCR models trained with the 12 different training datasets.



Number of Iteration

Figure 7. Performance comparison by the number of iterations for single LP-GANs and ensemble datasets.

As the last thing to focus on, the overall recognition performance with the Ensemble\_3k dataset was 95.56%, which is lower than 96.33% for pix2pix\_cGAN\_9k but higher than 93.59% for CycleGAN\_9k and 94.23% for StarGAN\_9k. Despite the number of training data items being three times higher, higher recognition performance was achieved when training with the 3000 training data items combining the LP images generated from multiple LP-GANs than training with a total of 9000 training data items generated from the single LP-GANs. The overall recognition performance was 98.72% for the Ensemble\_9k dataset with the 9000 combined data items, which is almost the same as when trained with real images. This shows that higher performance LPCR modules can be trained with LP images generated by multiple LP-GAN generators. Moreover, it is shown that the proposed LP-GAN can be fully used as a training data generator for the LPCR module of an ALPR system. Figure 7 shows the performance comparison of the ensemble datasets and the other single datasets.

Table 5 shows a comparison of the original YOLOv2 model and the proposed YOLOv2 model in terms of overall accuracy, average processing time, required GPU memory and number of floating-point operations (FLOPs) for the LPCR processes. Each model was trained with the same training dataset (Real\_9k) and tested with the same test dataset (Test\_13k). The original YOLOv2 model and the proposed YOLOv2 model were achieved 99.95%, 99.78% in overall accuracy, respectively. The proposed model is slightly less accurate than the original model. However, the proposed YOLOv2 model was modified the number of filters in the CNN layers, thereby the number of FLOPs was reduced from

Appl. Sci. 2020, 10, 2780 12 of 16

29.41 billion to 7.45 billion, and the processing time and the size of required GPU memory were also reduced in half (i.e., reducing the average processing time from 22 ms to 13 ms and the size of required GPU memory from 1006 MB to 474 MB).

	Overall Accuracy (%)	Average Processing Time (ms)	Required GPU Memory (MB)	Number of FLOPs (Bn)
Original YOLOv2	99.95	22	1006	29.41
Proposed YOLOv2	99.78	13	474	7.45

Table 5. Comparison of original YOLOv2 and proposed YOLOv2 model.

Figure 8 shows the results of the LPCR with the modified YOLOv2 model. It is difficult for LPCR using traditional segmentation-based ALPR methods to recognize the LP images because the performance was insufficient during the LP character segmentation stage due to distortion, contamination, illumination, and noise. However, since the modified YOLOv2 model detects the LP characters end-to-end without LP character segmentation (i.e., segmentation-free), its LPCR performance on the LP images was sufficient. Nevertheless, LPCR could not recognize the LP character information for some other LP images. Figure 9 shows some failure recognition resultants by the proposed YOLOv2 model. Although the proposed YOLOv2 model is robust to various environmental weaknesses, if the LP image is severely distorted due to excessive blurring or artificial manipulation, LPCR performance is degraded.



Figure 8. Some difficult recognition resultants of Korean LPs by proposed YOLOv2 model.



Figure 9. Some failure recognition resultants of Korean LPs by proposed YOLOv2 model.

#### 6. Conclusions

In this paper, we presented an LP image generator based on state-of-the-art GAN-based image-to-image translation methods to generate synthetic LP images using small set of real LP images for end-to-end LPCR module training. Our proposed LP-GAN generates LP images that are similar to the real ones using only the 159 real LP images available online. The generated synthetic images were used as the training data for the LPCR module with achieving 98.72% overall accuracy. Furthermore, the proposed method can be applied to generating of other countries' LP images as well as Korean ones. In addition, we presented the modified YOLOv2 model for an efficient LPCR module that performs character segmentation and recognition simultaneously in a segmentation-free end-to-end manner. Our proposed model was sped up 1.7 times faster than the original YOLOv2 model and the size of required GPU memory was also reduced in half.

**Author Contributions:** B.-G.H. conceived the idea and designed and performed the experiments. writing—original draft preparation, B.-G.H. and J.T.L.; writing—review and editing, K.-T.L. and D.-H.C. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean Government (Development of ICT Convergence Technology for Daegu-Gyeongbuk Regional Industry) under Grant 20ZD1100.

**Conflicts of Interest:** The authors declare no conflict of interest.

#### References

- 1. Du, S.; Ibrahim, M.; Shehata, M.; Badawy, W. Automatic license plate recognition (ALPR): A state-of-the-art review. *IEEE Trans. Circuits Syst. Video Technol.* **2012**, *23*, 311–325. [CrossRef]
- 2. Anagnostopoulos, C.N.E.; Anagnostopoulos, I.E.; Psoroulas, I.D.; Loumos, V.; Kayafas, E. License plate recognition from still images and video sequences: A survey. *IEEE Trans. Intell. Transp. Syst.* **2008**, *9*, 377–391. [CrossRef]
- 3. Lee, J.T.; Ryoo, M.S.; Riley, M.; Aggarwal, J. Real-time illegal parking detection in outdoor environments using 1-D transformation. *IEEE Trans. Circuits Syst. Video Technol.* **2009**, *19*, 1014–1024. [CrossRef]
- 4. Kim, K.J.; Kim, P.K.; Chung, Y.S.; Choi, D.H. Multi-Scale Detector for Accurate Vehicle Detection in Traffic Surveillance Data. *IEEE Access* **2019**, *7*, 78311–78319. [CrossRef]
- Weber, M. Caltech Cars Dataset. 1999. Available online: http://www.vision.caltech.edu/Image\_Datasets/ cars\_markus/cars\_markus.tar (accessed on 28 February 2020).

6. Laroca, R.; Severo, E.; Zanlorensi, L.A.; Oliveira, L.S.; Gonçalves, G.R.; Schwartz, W.R.; Menotti, D. A robust real-time automatic license plate recognition based on the YOLO detector. In Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), Rio, Brazil, 8–13 July 2018; pp. 1–10.

- 7. Srebrić, V. EnglishLP Database. 2003. Available online: http://www.zemris.fer.hr/projects/LicensePlates/english/baza\_slika.zip (accessed on 28 February 2020).
- 8. Dlagnekov, L.; Belongie, S. UCSD-Stills Dataset. 2005. Available online: http://vision.ucsd.edu/belongie-grp/research/carRec/car\_data.html (accessed on 28 February 2020).
- 9. Zhou, W.; Li, H.; Lu, Y.; Tian, Q. Principal visual word discovery for automatic license plate detection. *IEEE Trans. Image Process.* **2012**, 21, 4269–4279. [CrossRef] [PubMed]
- 10. Hsu, G.S.; Chen, J.C.; Chung, Y.Z. Application-oriented license plate recognition. *IEEE Trans. Veh. Technol.* **2012**, *62*, 552–561. [CrossRef]
- 11. OpenALPR Inc. OpenALPR-EU Dataset. 2016. Available online: https://github.com/openalpr/benchmarks/tree/master/endtoend/eu (accessed on 28 February 2020).
- 12. Gonçalves, G.R.; da Silva, S.P.G.; Menotti, D.; Schwartz, W.R. Benchmark for license plate character segmentation. *J. Electron. Imaging* **2016**, *25*, 053034. [CrossRef]
- 13. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; pp. 2672–2680.
- 14. Mirza, M.; Osindero, S. Conditional generative adversarial nets. arXiv 2014, arXiv:1411.1784.
- 15. Denton, E.L.; Chintala, S.; Szlam, A.; Fergus, R. Deep generative image models using a laplacian pyramid of adversarial networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 1486–1494.
- 16. Radford, A.; Metz, L.; Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv* **2015**, arXiv:1511.06434.
- 17. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein generative adversarial networks. In Proceedings of the International Conference on MACHINE Learning, Sydney, Australia, 6–11 August 2017; pp. 214–223.
- 18. Zhao, J.; Mathieu, M.; LeCun, Y. Energy-based generative adversarial network. arXiv 2016, arXiv:1609.03126.
- 19. Isola, P.; Zhu, J.Y.; Zhou, T.; Efros, A.A. Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1125–1134.
- Zhu, J.Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2223–2232.
- 21. Choi, Y.; Choi, M.; Kim, M.; Ha, J.W.; Kim, S.; Choo, J. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 8789–8797.
- 22. Kim, T.; Cha, M.; Kim, H.; Lee, J.K.; Kim, J. Learning to discover cross-domain relations with generative adversarial networks. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; Volume 70, pp. 1857–1865.
- 23. Ledig, C.; Theis, L.; Huszár, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.; Tejani, A.; Totz, J.; Wang, Z.; et al. Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4681–4690.
- 24. Johnson, J.; Alahi, A.; Fei-Fei, L. Perceptual losses for real-time style transfer and super-resolution. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 694–711.
- 25. Zhu, J.Y.; Krähenbühl, P.; Shechtman, E.; Efros, A.A. Generative visual manipulation on the natural image manifold. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 597–613.
- Hongliang, B.; Changping, L. A hybrid license plate extraction method based on edge statistics and morphology. In Proceedings of the 17th International Conference on Pattern Recognition, ICPR 2004, Cambridge, UK, 23–26 August 2004; Volume 2, pp. 831–834.

Appl. Sci. 2020, 10, 2780 15 of 16

27. Zheng, D.; Zhao, Y.; Wang, J. An efficient method of license plate location. *Pattern Recognit. Lett.* **2005**, 26, 2431–2438. [CrossRef]

- 28. Wu, H.H.P.; Chen, H.H.; Wu, R.J.; Shen, D.F. License plate extraction in low resolution video. In Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06), Hong Kong, China, 20–24 August 2006; Volume 1, pp. 824–827.
- 29. Xu, H.K.; Yu, F.H.; Jiao, J.H.; Song, H.S. A new approach of the vehicle license plate location. In Proceedings of the Sixth International Conference on Parallel and Distributed Computing Applications and Technologies (PDCAT'05), Dalian, China, 5–8 December 2005; pp. 1055–1057.
- 30. Lee, E.R.; Kim, P.K.; Kim, H.J. Automatic recognition of a car license plate using color image processing. In Proceedings of the 1st International Conference on Image Processing, Austin, TX, USA, 13–16 November 1994; Volume 2, pp. 301–305.
- 31. Matas, J.; Zimmermann, K. Unconstrained licence plate and text localization and recognition. In Proceedings of the 2005 IEEE Intelligent Transportation Systems, Vienna, Austria, 13–16 September 2005; pp. 225–230.
- 32. Han, B.G.; Lee, J.T.; Lim, K.T.; Chung, Y. Real-Time License Plate Detection in High-Resolution Videos Using Fastest Available Cascade Classifier and Core Patterns. *Etri J.* **2015**, *37*, 251–261. [CrossRef]
- 33. Kanayama, K.; Fujikawa, Y.; Fujimoto, K.; Horino, M. Development of vehicle-license number recognition system using real-time image processing and its application to travel-time measurement. In Proceedings of the 41st IEEE Vehicular Technology Conference, St. Louis, MO, USA, 19–21 May 1991; pp. 798–804.
- 34. Rahman, C.A.; Badawy, W.; Radmanesh, A. A real time vehicle's license plate recognition system. In Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance, Miami, FL, USA, 21–22 July 2003; pp. 163–166.
- 35. Guo, J.M.; Liu, Y.F. License plate localization and character segmentation with feedback self-learning and hybrid binarization techniques. *IEEE Trans. Veh. Technol.* **2008**, *57*, 1417–1424.
- 36. Capar, A.; Gokmen, M. Concurrent segmentation and recognition with shape-driven fast marching methods. In Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06), Hong Kong, China, 20–24 August 2006; Volume 1, pp. 155–158.
- 37. Yoon, Y.; Ban, K.D.; Yoon, H.; Lee, J.; Kim, J. Best combination of binarization methods for license plate character segmentation. *ETRI J.* **2013**, *35*, 491–500. [CrossRef]
- 38. Comelli, P.; Ferragina, P.; Granieri, M.N.; Stabile, F. Optical recognition of motor vehicle license plates. *IEEE Trans. Veh. Technol.* **1995**, 44, 790–799. [CrossRef]
- 39. Chang, S.L.; Chen, L.S.; Chung, Y.C.; Chen, S.W. Automatic license plate recognition. *IEEE Trans. Intell. Transp. Syst.* **2004**, *5*, 42–53. [CrossRef]
- 40. Türkyılmaz, İ.; Kaçan, K. License plate recognition system using artificial neural networks. *ETRI J.* **2017**, 39, 163–172. [CrossRef]
- 41. Yuan, Y.; Zou, W.; Zhao, Y.; Wang, X.; Hu, X.; Komodakis, N. A robust and efficient approach to license plate detection. *IEEE Trans. Image Process.* **2016**, *26*, 1102–1114. [CrossRef] [PubMed]
- 42. Meng, A.; Yang, W.; Xu, Z.; Huang, H.; Huang, L.; Ying, C. A robust and efficient method for license plate recognition. In Proceedings of the 24th International Conference on Pattern Recognition (ICPR), Beijing, China, 20–24 August 2018; pp. 1713–1718.
- 43. Špaňhel, J.; Sochor, J.; Juránek, R.; Herout, A.; Maršík, L.; Zemčík, P. Holistic recognition of low quality license plates by cnn using track annotated data. In Proceedings of the 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Lecce, Italy, 29 August–1 September 2017; pp. 1–6.
- Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation.
   In Proceedings of the International Conference on Medical image computing and computer-assisted intervention, Munich, Germany, 5–9 October 2015; Springer: Berlin/Heidelberg, Germany, 2015; pp. 234–241.
- 45. Demir, U.; Unal, G. Patch-based image inpainting with generative adversarial networks. *arXiv* **2018**, arXiv:1803.07422.
- 46. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
- 47. Mao, X.; Li, Q.; Xie, H.; Lau, R.Y.; Wang, Z.; Paul Smolley, S. Least squares generative adversarial networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2794–2802.

48. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.

- 49. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 91–99.
- 50. CycleGAN and pix2pix in PyTorch. Available online: https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix (accessed on 28 February 2020).
- 51. StarGAN. Available online: https://github.com/yunjey/stargan (accessed on 28 February 2020).
- 52. PyTorch org. Available online: https://pytorch.org (accessed on 28 February 2020).



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).