SNIDER: Single Noisy Image Denoising and Rectification for Improving License Plate Recognition

Younkwan Lee Juhyun Lee Hoyeon Ahn Moongu Jeon
Machine Learning and Vision Laboratory
Gwangju Institute of Science and Technology (GIST), Korea
{brightyoun, leejuhyun, ajhoyeon, mgjeon}@gist.ac.kr

Abstract

In this paper, we present an algorithm for real-world license plate recognition (LPR) from a low-quality image. Our method is built upon a framework that includes denoising and rectification, and each task is conducted by Convolutional Neural Networks. Existing denoising and rectification have been treated separately as a single network in previous research. In contrast to the previous work, we here propose an end-to-end trainable network for image recovery, Single Noisy Image DEnoising and Rectification (SNIDER), which focuses on solving both the problems jointly. It overcomes those obstacles by designing a novel network to address the denoising and rectification jointly. Moreover, we propose a way to leverage optimization with the auxiliary tasks for multi-task fitting and novel training losses. Extensive experiments on two challenging LPR datasets demonstrate the effectiveness of our proposed method in recovering the high-quality license plate image from the low-quality one and show that the the proposed method outperforms other state-of-the-art methods.

1. Introduction

License plate recognition (LPR) from the real-world is one of the fundamental problems in several intelligent transport systems (ITS) applications such as vehicle re-identification [22, 33], outdoor scene understanding [7, 25], and de-identification for privacy protection [10]. In the last few years, LPR has been widely studied in theoretical, experimental and numerical ways to provide robust image representation. Many LPR methods [2, 1, 11, 20] are capable of capturing the structural properties of images and noise for carefully constrained settings. Despite the recent success, recognizing license plate in the wild is still far from satisfactory due to the variations that suffer from appearance, noise, angle, and illumination.

Recently, due to the hierarchical feature extraction and learning capability, deep convolutional neural networks (CNNs) have made remarkable advances in many computer vision applications, such as object detection [30, 29], semantic segmentation [23, 31], action recognition [37], and face recognition [36, 27]. As a result, CNN-guided LPR methods are also extensively applied to handle the problem of recognizing license plate captured directly real-world camera. For example, Zhuang *et al.* [41] transform license plate into a semantic segmentation result with the counting network to handle appearance variations. Although numerous LPR methods have been developed [35, 41], they are not still capable of learning all types of samples in the wild. For this reasons, their algorithms practically assume a high-quality image as an input. Generally, the typical appearance of the license plate collected in real-world scenes might contain the aforementioned challenges, causing deterioration in LPR performance. Hence, developing and implementing robust LPR framework are highly indispensable, especially for real-world scenes.

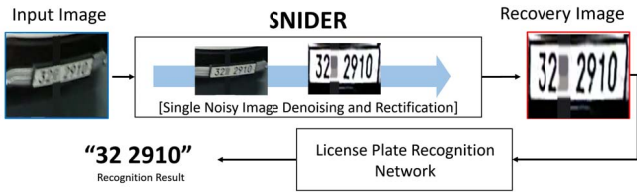

Figure 1. The proposed system consists of two components: single noisy image denoising and rectification (SNIDER) for recovering a low-quality license plate image and a license plate recognition (LPR) network for recognizing the final recovery image. The SNIDER is an end-to-end trainable network with auxiliary tasks for better image recovery. The LPR network uses a pre-trained DarkNet based on YOLO v3 to detect texts.

In this paper, we design an end-to-end single noisy image denoising and rectification network (SNIDER) for better LPR based on multiple auxiliary tasks. Figure 1 illustrates the LPR framework in which the proposed SNIDER is combined with a pre-trained LPR network. The SNIDER con-

2473-9944/19/$31.00 ©2019 IEEE
DOI 10.1109/ICCVW.2019.00131

sists of two sub-networks: a denoising network and a rectification network. Motivated by the success of U-Net [31] in recovering the object details, we employ U-Net structure as an image recovery backbone network, attempting to extract visual content at structural-level details. In the denoising sub-network (DSN), we try to transform a low-quality image to a high-quality image pixel by pixel directly. The DSN can penalize the loss between noisy and noise-free image pairs and thus acquire the output image with the fine textures of the clean component, learning an independent realization of the noise. However, even with such sophisticated DSN, denoising images are unsatisfactory because they still have arbitrary geometric variations. Therefore, the rectification sub-network (RSN) is proposed to correct geometric distortions of denoising license plates and generate more accurate correction image distortion. Furthermore, we propose to leverage the new auxiliary tasks to further optimize the image recovery sub-networks (DSN, RSN) of SNIDER. There are two auxiliary tasks: a text counting module and a segment prediction module. Specifically, we solve each auxiliary module using CNN as a decoder. The counting module is used to predict the number of text in the image as a classification problem. In this module, despite the ambiguous boundary of consecutive text, text counting can distinguish single text, which makes the image quality suitable for text detection. For the segment prediction module, we propose a binary segmentation to emphasize the foreground over the background. The generated segmentation result makes the license plate clean for text recognition. Finally, learning the auxiliary tasks will lead the intermediate features of the recovery main task networks to enhance the difficulties such as geometric variations and low-quality information. More importantly, we introduce a new loss function that trains the SNIDER with auxiliary tasks, which provide significantly higher license plate quality for robust LPR.

To sum up, we highlight the main contributions of this paper as follows:

- We propose a novel end-to-end license plate recovery network, where denoising and rectification network are used to generate a clear recovery image for robust LPR performance.
- We present the auxiliary tasks to leverage the quality of the license plate recovery from low-quality. Mainly a new loss is introduced to provide regularization effects to the backbone SNIDER for robust representation and license plate recovery.
- Finally, we demonstrate the effectiveness of the proposed method in recovering a high-quality license plate from a low-quality license plate in the real-world and show that the LPR performance outperforms the state-of-the-art methods on two challenging datasets, AOLP-RP [13] and VTLPs dataset newly collected on the most challenging real-world environments.

2. Related Work

In this section, we briefly review on low-quality image recovery methods and license plate recognition methods that is most related to this work.

2.1. Low-Quality Image Recovery

To obtain the high-quality image, most of the existing methods depend on the assumption that both signal and noise arise from particular statistical regularities by using hand-crafted algorithms, such as anisotropic diffusion [28] and total variation [32]. Besides, non-parametric models [8, 26] were developed to model image noise, but they were also not robust to the unconstrained environment in the wild due to priors estimated from limited observations. Recently, due to the advances in deep learning, most denoising algorithms are designed with deep neural network architectures and data-driven approach rather than relying on the priors. Burger *et al.* [3] employ multi-layer perceptrons with a data-driven technique based on an extensive image database. Zhang *et al.* [38] train the deep CNN by utilizing batch normalization (BN) [14] and residual learning [12].

Though useful for estimating a clean image, text classifiers are still hard to recognize due to the irregular text geometry. It motivates research for image recovery to extend image rectification. Shi *et al.* [34] develop a spatial transformer network (STN) for rectifying text distortion. Cheng *et al.* [6] adopt more in-depth representations of images by a residual network. Different from the existing methods, in this paper, we extract deep representations of images using the U-Net-based CNN for denoising as well as rectification. To the best of our knowledge, our research may be first work to apply the two modules mentioned above for LPR at the same time.

2.2. License Plate Recognition

Before the advent of deep learning, most of the traditional LPR methods [16, 1, 13, 40] employ two-stage process flow, involving text detection and following text recognition. After the advancement of deep learning, many approaches employ a one-stage process flow without text detection. Li *et al.* [20] extract deep feature representations by using RNN with LSTM for acquiring sequential features of the license plate. Bulan *et al.* [2] estimate domain shifts between target and multiple source domains for selecting a domain that yields the best recognition performance based on fully convolutional network [23]. However, these methods only consider high-quality license plate image except for low-quality image, which is easily led to low performance in real-world scenes. Moreover, their methods lack little or

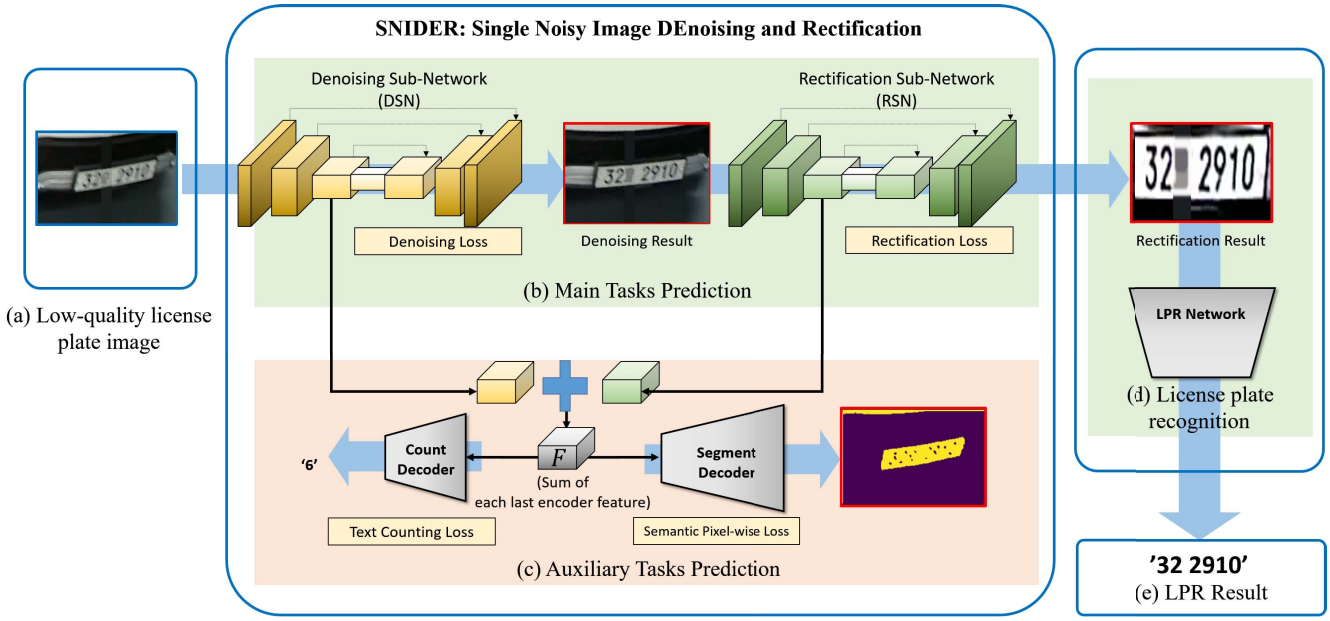

Figure 2. The training and testing process of the proposed approach with the learning of two auxiliary tasks: (a) The input images are fed into SNIDER for the image recovery; (b, c) SNIDER consists of main tasks (*i.e.* DSN, RSN) and auxiliary tasks, they transform low-quality data into high-quality data for training the DSN, RSN and auxiliary tasks networks; (d, e) LPR network is for testing and outputs a LPR result. The DSN is trained to generate a denoising image from the low-quality input image. Also, the RSN is trained to generate a rectified image from the result of DSN. The auxiliary tasks include text counting and binary segmentation, which are formulated as classification and regression simultaneously.

no effort to improve image quality, while requiring a lot of computing power. In this work, unlike existing methods, we adopt image recovery for high LPR performance under the low-quality image in real-world scenes. To the best of our knowledge, this is the first time we apply sophisticated image recovery to handle a challenging real-world environment. Besides, our methods are computationally efficient and capable of real-time recognition despite additional recovery modules.

3. Proposed Method

The proposed approach consists of three parts: 1) main tasks prediction networks G_D and G_R for denoising and rectification; 2) auxiliary tasks prediction networks D_c and D_s for count classification and segment prediction; 3) LPR network for text detection and classification. The proposed architecture is illustrated in Figure 2. For training, dataset for main tasks and auxiliary tasks can be inferred from the intentionally transform operation by simple rotation (for rectification) and down-resizing (for denoising), as shown in Figure 3. In particular, only one sample of original image I^{HQ} simply can generate four training samples that have been transformed by different angles. Given the training samples I_i^{HQ} for G_D, I_i^{LQ} for G_R, I_i^{seg} for D_s and c for D_c, $i \in \{-30\,^\circ, -15\,^\circ, +15\,^\circ, +30\,^\circ\}$, the main tasks

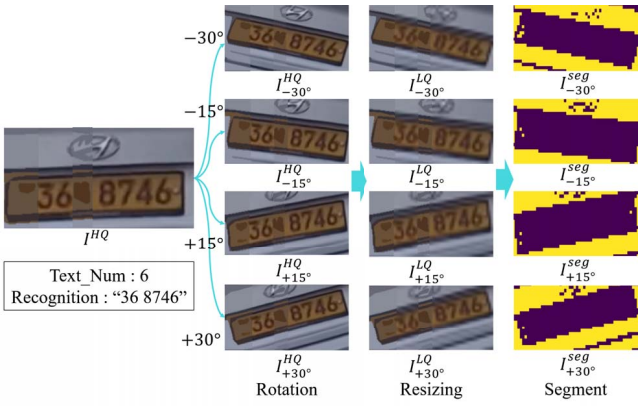

Figure 3. Label generation for the training of the proposed method. From a high-quality image as ground truth, the rotated images can be obtained using a simple linear transformation, and the low-quality image is processed through downsampling $\times$ 1/4 of rotation image. The segmented image is inferred through binarization [4] of the low-quality image.

G_D and G_R extract recovery result from input image I_i^{LQ} and corresponding samples. LPR network LPR then takes $G_R(G_D(I_i^{LQ}))$ to recognize a recovery image.

In the following subsections, we introduce the method to predict the main tasks in Section 3.1. Then, we also address

the auxiliary tasks for prediction in Section 3.2. Then, we describe the network training of the proposed architecture in Section 3.3. Finally, we illustrate the testing process in Section 3.4.

3.1. Denoising and Rectification Network

Our main task networks include two sub-networks (*i.e.* denoising sub-network and rectification sub-network), and the first sub-network takes the low-quality image as the input, and the output is the recovered image. In this paper, we design the rectification network to rectify the denoising results from the denoising network.

The image recovery results [15] have shown the effectiveness of the U-Net since it can provide high-quality overall details of an image object, without a negative impact on the image generation. Therefore, we adopt a U-Net-based architecture adding skip connections that shuffle low-level information shared between input and output across the network. In contrast to their network, our recovery network includes two sub-networks, which are also the U-Net architecture. As shown in Table 1 and Figure 2.(b,c), our denoising network G_D and rectification network G_R consist of the encoder and decoder module.

To achieve the main tasks, we first feed I_i^{LQ} into G_D to generate denoising results. Given a pair of input image and non-rectified ground-truth denoising image $\{I_{i,j}^{LQ}, I_{i,j}^{HQ}\}_{(i,j)}^{N}$, loss function for the G_D is the pixel-wise MSE loss, and it is calculated as Eq. (1):

$$\mathcal{L}_{G_D}(w) = \frac{1}{N} \sum_{(i,j)\in N}^{N} \parallel G_{D_w}(I_{(i,j)}^{LQ}) - I_{(i,j)}^{HQ} \parallel^2, \quad (1)$$

where w is the parameters of denoising network. Such loss function encourages the G_D to not only extract the content information of input image but also generate a high-quality natural image in pixel level.

Then, the rectification sub-network G_R processes the output from G_D, and outputs a rectified high-quality image, which is easier for the LPR network to recognize the identification text. With the training pairs of $\{G_D(I_{i,j}^{LQ}), I_{i=0,j}^{HQ}\}_{(i,j)}^{N}$, the G_R can be trained using a L1 loss for the predicted result $G_R(G_D(I_{i,j}^{LQ}))$:

$$\mathcal{L}_{G_R}(w) = \frac{1}{N} \sum_{(i,j)\in N}^{N} \parallel G_{R_w}(G_{D_w}(I_{(i,j)}^{LQ})) - I_{(i=0,j)}^{HQ} \parallel^1, \quad (2)$$

where w is the parameters of the rectification network. Unlike L2 loss, using L1 loss in the pixel level helps to preserve the appearance of an object, such as image color, intensity, and illumination, and leads to denoising result capable of only geometric transformation. Therefore, we can only perform geometric transformations without the appearance damage of the image during the rectification process, which forces the recognizer to be helpful.

Table 1. Details of different proposed network architectures.

	SNIDER	SNIDER-Tiny
Encoder (for G_D, G_R)	conv1(1,2) : (3×3×32 conv)×2, stride 2 pool1 : (2×2) max pooling, stride 2 conv2(1,2) : (3×3×64 conv)×2, stride 2 pool2 : (2×2) max pooling, stride 2 conv3(1,2) : (3×3×128 conv)×2, stride 2 pool3 : (2×2) max pooling, stride 2 conv4(1,2) : (3×3×256 conv)×2, stride 2 pool4 : (2×2) max pooling, stride 2 conv5(1,2) : (3×3×512 conv)×2, stride 2	7×7×32 conv, stride 2 7×7×64 conv, stride 2 5×5×128 conv
Decoder (for G_D, G_R, D_s)	x2 upsample : concat(conv5_2, conv4_2) conv6(1,2) : (3×3×256), stride 2 x2 upsample : concat(conv6_2, conv3_2) conv7(1,2) : (3×3×128), stride 2 x2 upsample : concat(conv7_2, conv2_2) conv8(1,2) : (3×3×64), stride 2 x2 upsample : concat(conv8_2, conv2_2) conv9(1,2) : (3×3×32), stride 2 conv10 : (1×1×3)	5×5×64 conv 7×7×32 deconv, dilate 2 7×7×3 deconv, dilate 2
Decoder (for D_c)	N×N×512 conv 1×1×256 conv 1×1×128 conv 1×1×64 conv 1×1×1 conv	N×N×128 conv 1×1×64 conv 1×1×1 conv

3.2. Auxiliary Tasks Prediction

Due to the complex real-world environments such as the extremely irregular geometric shape of text as well as the complicated image background, the binary information of the license plate is often noisy. Although we intend G_D and G_R to capture robust features for image recovery, the results by this structure do not always guarantee a well-enhanced output. Therefore, our work involves an additional learning branch where a richer feature representation is obtained from the backbone network. Motivated by multi-task learning [5], we employ the auxiliary tasks, *i.e.*, binary segmentation and count estimation, which will contribute our main task networks produce more discriminative feature representations. Towards this problem, we sum the weights of the last layer of encoders in order to guide auxiliary task networks to help main task networks effectively extract critical information from the low-quality image.

For the binary segmentation task, we introduce the segment decoder D_s based on U-Net architecture. Detailed architectures of the D_s are shown in Table 1. The D_s accepts feature set F summed from the last features of each main task's encoder and outputs a license plate segment with val-

ues indicating the probability of pixels belonging to the license plate. Also, ground-truth labels for segmentation can be inferred from the dotted annotations by [4]'s method as Otsu Thresholding, as shown in Figure 3. Although our segmentation annotations by [4] do not fully reflect the actual detail appearance of an image, we have shown in the experiments that this auxiliary and straightforward learning strategy leads to effective advances in image recovery. Given a pair of F and the ground-truth segmentation result in I^{seg}, loss function for the D_s is the binary cross-entropy loss:

$$\begin{aligned}\mathcal{L}_{D_s}(w) = \frac{1}{N}\sum_{(x,y)\in N}^{N} I^{seg}_{(x,y)} log(D_{s_w}(F)_{(x,y)}) \\ +(1-I^{seg}_{(x,y)}) log(1-D_{s_w}(F)_{(x,y)}),\end{aligned} \tag{3}$$

where $I^{seg}_{(x,y)} \in \{0,1\}$ is the real classes of pixels in I^{seg} with 1 for the license plate area and 0 for the background, $D_s(F)_{(x,y)}$ denotes the pixel-wise probability by D_s.

Also, we find that the generated recovery samples cannot usually distinguish successive texts due to close to each other. Motivated by the observations, we add a counting decoder D_c, which predicts the number of characters in the image. As a result, our D_c plays two roles, where the first is to cause separation between adjacent texts more clearly. The other role is to promote the encoders of each main task to generate a higher quality image while backpropagating the penalty. The loss function for the D_c is the L2 loss:

$$\mathcal{L}_{D_c} = \parallel C_{pred} - C_{G.T} \parallel^2, \tag{4}$$

where C_{pred} and $C_{G.T}$ are the predicted value and the ground-truth, respectively.

3.3. Network Training

The full objective function is a weighted sum of all the losses from Eq. (1) to (4):

$$\mathcal{L} = \lambda_{G_D}\mathcal{L}_{G_D} + \lambda_{G_R}\mathcal{L}_{G_R} + \lambda_{D_s}\mathcal{L}_{D_s} + \lambda_{D_c}\mathcal{L}_{D_C} \tag{5}$$

We employ a stage-wise training strategy to optimize main tasks with auxiliary tasks and empirically set the weights of each loss as detailed in Section 5.3.

3.4. Testing

At the testing phase, the auxiliary tasks are removed. Given a low-quality test image I_{test}, G_D and G_R output the recovered image via denoising and rectification. Then LPR network LPR based on a YOLO v3 detector [29] by pre-trained on ImageNet [9] takes the recovered image and generates the recognition result LPR_{result} of I_{test}, and it is denoted as Eq. (6):

$$LPR_{result} = LPR(G_R(G_D(I_{test}))). \tag{6}$$

4. Experimental Setting

In this section, we describe a list of datasets, metric, and implementation details for the proposed method.

4.1. Datasets

We use LPR datasets AOLP [13] and newly collected dataset, named VTLP.

AOLP-RP : AOLP-RP [13] consists of 611 images collected in Taiwan, including ten numbers and 25 letters (except "O"). This dataset has a challenging factor that the angle of the LP contains oblique samples in terms of distortion. On the other hand, in terms of resolution, all images are relatively easy because they consist of high-resolution samples rather than other datasets.

VTLP : We introduce a new challenging large-scale dataset collected in South Korea. The dataset contains 10,650 LP images, which are divided into 6,400/4,250 images for training and testing, respectively. All Korean letters are hidden for privacy protection. Images in VTLP consist of text(only 10 digits, not Korean). Compared with the public LPR datasets, our dataset has challenging factors: 1) We apply the manual annotation of large-scale images selected from unconstrained real-world, covering a variety of challenging situations using bounding box coordination; 2) Distance from vehicles to the camera is far from other dataset; 3) Various scene-texts interfere with the detection, low-resolution appearance, and very oblique LP.

4.2. Evaluation Metric

We follow the evaluation metric that has been widely used in LPR research [13, 41]. Therefore, if only one of the consecutive characters is misclassified or not detected, it is treated as a failure case. We denote this metric as a recognition accuracy. Also, we address the 36 characters, including 26 letters and 10 digits for text recognition.

4.3. Implementation Details

All the reported implementations are based on the TensorFlow framework, and our method has done on one NVIDIA TITAN X GPU and one Intel Core i7-6700K CPU. In all the experiments, we resize all images to 320 $\times$ 320. For stable training, we use a gradient clipping trick and the Adam optimizer [17] with high momentum. The proposed network is trained in 1 million iterations with a batch size of 16. The weights in all SNIDER layers are initialized from a zero-mean Gaussian distribution with a standard deviation of 0.01, and the constant 0 as the biases in all layers. All models are trained for the first 100 epochs with a learning rate of 10^{-4} despite higher values, and then for the remaining epochs at the learning rate of 10^{-5}. Batch normalization [14] and LeakyReLU [24] are used in all layers of our networks. Also, for LPR network as baseline, we use the YOLO v3 detector [29] model pre-trained on ImageNet[9].

(a) (b) (c) (d) (e) (f) (g)

Figure 4. Ablation Study. (a) : shows noisy input; (b) : only contains denoising net; (c) : only contains rectification net; (d) : adds all main tasks; (e) : adds segment task from (d); (f) : adds counting task from (d); (g) : adds all of tasks, namely our proposed model.

Table 2. Ablation study on the effectiveness of different components. DSN, RSN, SD, and CD represent the G_D, G_R, D_s, and D_c , respectively.

Type	Method	LPR Accuracy	
		AOLP	VTLP
a	Baseline (YOLO v3)	91.65	80.45
b	Add DSN	91.98	84.64
	Add RSN	97.05	87.13
c	Add DSN, RSN	98.53	90.71
d	Add DSN, RSN, SD	99.02	92.08
	Add DSN, RSN, CD	98.69	91.08
e	Add DSN, RSN, SD, CD (ours)	**99.18**	**93.08**

Table 3. Full LPR performance (percentage) comparison of our method with the existing methods on **AOLP-RP [13].**

Method	AOLP-RP Full LPR accuracy (%)
Baseline (YOLO v3)	91.65
Hsu *et al.* [13]	85.76
Li *et al.* [21]	88.38
Silva *et al.* [35]	98.36
Zhuang *et al.* [41]	99.02
SNIDER-Tiny	**98.85**
SNIDER	**99.18**

Table 4. Full LPR performance (percentage) comparison of our method with the existing methods on **VTLP**.

Method	VTLP Full LPR accuracy (%)
Baseline (YOLO v3)	80.45
Laroca *et al.* [18]	87.34
Silva *et al.* [35]	84.73
SNIDER-Tiny	86.66
SNIDER	**93.08**

Two SNIDER models are trained for evaluations and benchmarking with state-of-the-art methods. The first is a backbone model **SNIDER**, which uses five convolution blocks at encoder and decoder, respectively. In contrast, the other model denoted by **SNIDER-Tiny** uses a relatively light network thereby too fast for testing. All SNIDER models are trained under the same parameter setting.

5. Results

In this section, we evaluate the proposed approach on two datasets: AOLP-RP [13] and VTLP.

5.1. Ablation Study

We first compare our proposed method with the baseline LPR network to prove the effectiveness of image recovery performance. Both LPR results on two datasets are reported for the following five types of our methods where each module is optionally added: a) the only baseline without proposed method; b) adding one main task; c) adding all main tasks; d) adding all main tasks and one auxiliary task; e) adding all of the modules (namely, proposed method).

We present the LPR accuracy for each type on two datasets in Table 2, and the visual comparisons are shown in Figure 4. From Table 2, we can find that adding the denoising and the rectification task, respectively, significantly improves the LPR performance (type b, c). In addition, we observe that LPR performance improves more when both tasks are applied at the same time. As shown in Figure 4. (c), noise and blurring effect are removed from the low-quality image (a), and characters are enhanced well compared to (c). This confirms that performing two tasks at the same time is more helpful to recover high-quality images. Despite showing better LPR performance (Table 2. (c)), we still find that the output image contains elements that interfere with LPR performance. For example, there are still challenges to detect the suitable text region, including a region that is unnecessary for recognition, such as a manufacturer's logo (see in Figure 4. (d)), and ambiguity that not well detected between consecutive characters. Therefore, when each auxiliary task is added to main tasks, recovered image quality can be better (Figure 4. (e,f)) and we observe some improvements on LPR performance (Table 2. (d)). Finally, we incorporate all the tasks, perform experiments on it and observe the best performance improvement in LPR (Table 2. (e)). Furthermore, the recovered image in Figure 4. (g) is the most realistic of all results.

Figure 5. Visual comparison of different license plate recovery methods: For the three sample images in the first column, columns 2-4 show the recovery images by using [34], [38] and SNIDER, respectively. The sample images are from VTLP which suffer from geometric distortions as well as low-quality. The proposed SNIDER performs better in LPR recovery. Best viewed on the computer, in color and zoomed in.

5.2. Comparison with State-of-the-art Methods

We compare the proposed method with some state-of-the-art LPR methods [13, 21, 35, 41]. For the baseline LPR, the SNIDER has been evaluated over the two datasets as described in Section 4.1 that contain low-quality license plate images with a variety of geometric variations.

As Table 2, 3 and 4 show, the **SNIDER** consistently outperforms the **SNIDER-Tiny** across all datasets due to the use of a more in-depth and broader backbone network. However, **SNIDER-Tiny** is also evaluated to be more effective than most methods, and if not, it shows a relatively small performance difference. Therefore, it can be explained that SNIDER is more useful for LPR than other methods for the low-quality image.

AOLP-RP dataset results. For the AOLP-RP, SNIDER demonstrates that our recovery image can significantly improve the performance of LPR on real-world images. This is mainly due to the fact that AOLP dataset which usually have geometrically tilted cases is processed into a well-rectified image. The results are listed in Table 3, and our method obtains the highest performance (**99.18%**), and outperforms the state-of-the-art LPR methods by more than 0.16%. Note that what we want to illustrate in the AOLP-RP evaluation (especially see the difference between Baseline and ours in Table 3) is that our method can benefit from the SNIDER, which enhances the image quality despite oblique angle.

VTLP dataset results. The quantitative results for VTLP dataset are shown in Table 4 and the visual comparisons are illustrated in Figure 5. Our approach shows superior performance to other LPR algorithms on LPR accuracy and image recovery. Furthermore, we achieve comparable results with state-of-the-art LPR method [18, 35]. From Table 4, our method obtains the highest performance (**93.08%**), and outperforms the state-of-the-art methods by more than 5.74% (87.34% *vs* 93.08%). Note that SNIDER achieves robust performance in VTLP that are collected in low-resolution environments rather than other datasets.

5.3. Parameter Study of the Weights for Tasks

The set of weights λ in Eq.(5) determines the influence of each task. To choose the optimal selection of λ, we perform various experiments with the SNIDER model on AOLP-RP and VTLP dataset. Since the influence of the main task is larger than that of the auxiliary task, the weight is also set higher. We also need to adjust the weights for fast optimization even within the auxiliary task. Figure 4 shows the segment decoder D_s plays an important role in eliminating unnecessary areas that interfere with LPR. Therefore, we set the weight of the segment decoder higher than the counting decoder. In our experiment, we set the weights for λ_{G_D}, λ_{G_R}, λ_{Ds} and λ_{Dc} to 0.4, 0.4, 0.15, and 0.05, respectively.

Image	SNIDER	G.T.	Predicted	Cause of Failure
AOLP-RP Dataset				
		CN 9139	N 9139	'CN' are incompletely rectified, LPR network makes a mistake in detection
		9863 QX	8863 QX	The entire image is incompletely rectified, LPR network makes a mistake in classification
VTLP Dataset				
		46 0507	46 0607	Extremely noisy condition, LPR network makes a mistake in classification
		19 6824	19 6874	'6824' are incompletely rectified, LPR network makes a mistake in classification
		57 5448	57 59448	The entire image is incompletely rectified, LPR network makes a mistake in detecting
		33 1816	33 7876	The entire image is incompletely rectified, LPR network makes a mistake in classification
		11 0199	11 0198	The entire image is incompletely rectified, LPR network makes a mistake in classification

Figure 6. Error study on AOLP and VTLP dataset. Best viewed on the computer, in color and zoomed in.

Table 5. Impact of improving the LPR network and its performance evaluation with SNIDER for the VTLP testing set.

Baseline	LPR Accuracy	FPS
Faster R-CNN [30]	87.06	2.7
CornerNet-Squeeze [19]	93.39	13.1
CenterNet ResNet-18 [39]	84.68	46
YOLO v3 [29] (SNIDER-Tiny)	86.66	44
YOLO v3 [29] (ours)	93.08	37

5.4. Impact of LPR Network

We evaluate how LPR network choice impact LPR performance on the VTLP testing set. Results are shown in Table 5. We mainly adopt a real-time detector for fast processing. Compare with [30, 39], SNIDER indicates that the detector plays an important role in LPR performance. Although previous detectors are high-speed processing through lightweight models, they do not guarantee accuracy. Thus, we adopt YOLO v3, which corresponds to the adequate model that includes enough capacity for rich feature representation during real-time processing.

5.5. Weakness Analysis

Figure 6 shows some failure cases, including some false recovery results. These results identify that more progress is needed to improve the rectification performance further. Future work will address this problem by adding the adjacent context to recovering these more challenging license plate images.

6. Conclusion

In this paper, we propose a new end-to-end trainable image recovery method that is capable of recognizing license plates in the real-world. The proposed recovery network consists of two sub-networks, the denoising sub-network and the rectification network. In particular, two auxiliary tasks are designed to leverage the recovery of license plates, promoting the feature set to be more robust against the geometric variations and blurry data in the real-world scenes. Moreover, a new loss function is introduced to the backbone network to provide regularization effects and a higher-quality recovery image. Extensive experiments over various datasets demonstrate superior performance in license plate recovery and recognition.

Acknowledgement

This work was supported by Institute of Information communications Technology Planning Evaluation (IITP) grant funded by the Korean government (MSIT) (No.B0101-16-0525, development of global multi-target tracking and event prediction techniques based on real-time large-scale video analysis. We also appreciate useful discussions with Kyungho Won, Jaewoong Yun, and Sangwoo Park.

References

[1] 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*, 7(3):377–392, 2006.

[2] O. Bulan, V. Kozitsky, P. Ramesh, and M. Shreve. Segmentation-and annotation-free license plate recognition with deep localization and failure identification. *IEEE Transactions on Intelligent Transportation Systems*, 18(9):2351–2363, 2017.

[3] H. C. Burger, C. J. Schuler, and S. Harmeling. Image denoising: Can plain neural networks compete with bm3d? In *2012 IEEE conference on computer vision and pattern recognition*, pages 2392–2399. IEEE, 2012.

[4] H. Cai, Z. Yang, X. Cao, W. Xia, and X. Xu. A new iterative triclass thresholding technique in image segmentation. *IEEE transactions on image processing*, 23(3):1038–1046, 2014.

[5] R. Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

[6] Z. Cheng, F. Bai, Y. Xu, G. Zheng, S. Pu, and S. Zhou. Focusing attention: Towards accurate text recognition in natural images. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[7] S. Cherng, C.-Y. Fang, C.-P. Chen, and S.-W. Chen. Critical motion detection of nearby moving vehicles in a vision-based driver-assistance system. *IEEE Transactions on Intelligent Transportation Systems*, 10(1):70–82, 2009.

[8] K. Dabov, A. Foi, and K. Egiazarian. Video denoising by sparse 3d transform-domain collaborative filtering. In *2007 15th European Signal Processing Conference*, pages 145–149. IEEE, 2007.

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[10] L. Du and H. Ling. Preservative license plate de-identification for privacy protection. In *2011 International Conference on Document Analysis and Recognition*, pages 468–472. IEEE, 2011.

[11] C. Gou, K. Wang, Y. Yao, and Z. Li. Vehicle license plate recognition based on extremal regions and restricted boltzmann machines. *IEEE Transactions on Intelligent Transportation Systems*, 17(4):1096–1107, 2015.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[13] G.-S. Hsu, J.-C. Chen, and Y.-Z. Chung. Application-oriented license plate recognition. *IEEE transactions on vehicular technology*, 62(2):552–561, 2012.

[14] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[15] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[16] K. K. Kim, K. Kim, J. Kim, and H. J. Kim. Learning-based approach for license plate recognition. In *Neural Networks for Signal Processing X. Proceedings of the 2000 IEEE Signal Processing Society Workshop (Cat. No. 00TH8501)*, volume 2, pages 614–623. IEEE, 2000.

[17] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[18] R. Laroca, E. Severo, L. A. Zanlorensi, L. S. Oliveira, G. R. Gonçalves, W. R. Schwartz, and D. Menotti. A robust real-time automatic license plate recognition based on the yolo detector. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–10. IEEE, 2018.

[19] H. Law, Y. Teng, O. Russakovsky, and J. Deng. Cornernet-lite: Efficient keypoint based object detection. *arXiv preprint arXiv:1904.08900*, 2019.

[20] H. Li and C. Shen. Reading car license plates using deep convolutional neural networks and lstms. *arXiv preprint arXiv:1601.05610*, 2016.

[21] H. Li, P. Wang, and C. Shen. Toward end-to-end car license plate detection and recognition with deep neural networks. *IEEE Transactions on Intelligent Transportation Systems*, 20(3):1126–1136, 2018.

[22] H. Liu, Y. Tian, Y. Yang, L. Pang, and T. Huang. Deep relative distance learning: Tell the difference between similar vehicles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2167–2175, 2016.

[23] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[24] A. L. Maas, A. Y. Hannun, and A. Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3, 2013.

[25] S. Noh and M. Jeon. A new framework for background subtraction using multiple cues. In *Asian Conference on Computer Vision*, pages 493–506. Springer, 2012.

[26] J. Pan, D. Sun, H. Pfister, and M.-H. Yang. Blind image deblurring using dark channel prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1628–1636, 2016.

[27] S. Park, J. Yu, and M. Jeon. Learning feature representation for face verification. In *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6. IEEE.

[28] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on pattern analysis and machine intelligence*, 12(7):629–639, 1990.

[29] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[30] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[31] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[32] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992.

[33] Y. Shen, T. Xiao, H. Li, S. Yi, and X. Wang. Learning deep neural networks for vehicle re-id with visual-spatio-temporal path proposals. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1900–1909, 2017.

[34] B. Shi, X. Wang, P. Lyu, C. Yao, and X. Bai. Robust scene text recognition with automatic rectification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4168–4176, 2016.

[35] S. M. Silva and C. R. Jung. License plate detection and recognition in unconstrained scenarios. In *European Conference on Computer Vision*, pages 593–609. Springer, 2018.

[36] Y. Wen, K. Zhang, Z. Li, and Y. Qiao. A discriminative feature learning approach for deep face recognition. In *European conference on computer vision*, pages 499–515. Springer, 2016.

[37] J. Yu, S. Park, S. Lee, and M. Jeon. Driver drowsiness detection using condition-adaptive representation learning framework. *IEEE Transactions on Intelligent Transportation Systems*, 2018.

[38] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017.

[39] X. Zhou, D. Wang, and P. Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019.

[40] S. Zhu, S. A. Dianat, and L. K. Mestha. End-to-end system of license plate localization and recognition. *Journal of Electronic Imaging*, 24(2):023020, 2015.

[41] J. Zhuang, S. Hou, Z. Wang, and Z.-J. Zha. Towards human-level license plate recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 306–321, 2018.