License plate recognition in low quality image by using Latent Diffusion YOLOv7

*Yu-Hsi Chen
Institute of Information Science (IIS)
Academia Sinica
Taipei, Taiwan
franktpmvu@gmail.com

*Cheng-Jung Chuang
Institute of Information Science (IIS)
Academia Sinica
Taipei, Taiwan
candy0016@iis.sinica.edu.tw

Chien-Yao Wang
Institute of Information Science (IIS)
Academia Sinica
Taipei, Taiwan
kinyiu@iis.sinica.edu.tw

Jen-Chun Lin
Institute of Information Science (IIS)
Academia Sinica
Taipei, Taiwan
jenchunlin@iis.sinica.edu.tw

Hong-Yuan Mark Liao
Institute of Information Science (IIS)
Academia Sinica
Taipei, Taiwan
liao@iis.sinica.edu.tw

***Abstract*—The biggest challenge focus by today's license plate recognition systems is whether they can still operate normally when weather conditions are severe. Many current systems rely on deep learning methods to train the system, however, collecting data under severe weather often requires a lot of costs. Because the quality of collected data is generally poor, which will greatly increase the difficulty of manual annotation. In addition, since regional license plates vary greatly, it will take a lot of effort to collect plate data in different regional styles in order to develop a universally-applicable recognition system.**

This paper mainly explores the problem of rapid performance degradation of existing license plate recognition systems under harsh weather conditions. We propose a method that can simultaneously generate fully synthetic license plate data as well as noises under severe weather conditions. Furthermore, we designed a new Latent Diffusion YOLOv7 (LD-YOLOv7) neural network based on the existing object detection method—YOLOv7, which can effectively solve various problems of plate recognition in bad weather. In order to verify the effectiveness of this system in practical application, we also established a regional license plate dataset containing real harsh climate conditions for testing purposes. The experimental results show that our proposed model performs comparable to other existing methods trained with real data on AOLP dataset, and performs better under harsh weather conditions.

***Index Terms*—license plate recognition, object detection, latent diffusion, synthetic data.**

I. Introduction

Automatic License Plate Recognition (ALPR) is a very important and practical computer vision technology in modern life. Its applications can be seen everywhere in smart parking lots, electronic road toll collection, or access control systems. The early ALPR system needed to limit the direction and position of the car during use and count on bright light to guarantee effective function [5]. In recent years, thanks to the rapid development of deep learning technology, researchers have the opportunity to focus on more difficult outdoor scenes. However, systems based on deep learning rely heavily on whether the dataset is completely collected, so there are studies such as AOLP [9], CCPD [30], and CRPD [6], trying to build real-world outdoor scene datasets. In addition, Azam et al. [2] analyzed various factors that affect the accuracy of the license plate recognition system in real outdoor scenes, such as low brightness, bad weather, etc.

Under harsh weather conditions, the two most difficult problems that general deep learning-based license plate recognition systems need to overcome are the issues of insufficient training data and noise. Because real data in severe weather is difficult to collect, it becomes imperative to use synthesis as a means of data augmentation. The synthesized data does not involve the two most labor-intensive tasks of labeling and data collection, and the angle of the license plate and the text and numbers on the license plate can be adjusted at will, which is very beneficial to the development of the system. Wrenninge and Unger [29] believe that making synthetic data closer to real conditions can improve the system's recognition capabilities, so they recommend using rendering technology or game engines to construct realistic urban scenes, and they have achieved considerable results. However, the above methods often ignore the complex calculations of high-fidelity rendering and the large costs required for game modeling. Since a license plate has a clear appearance definition, there is no need to synthesize too realistic scenes. As long as the synthesized data conforms to the actual license plate rules and is given enough randomness, good results can be achieved.

Based on the premise of using synthetic augmentation datasets, we propose two methods to solve the problems caused by noise. The first one is to use a noise generator and add weather noise to the synthetic data. This can adapt the network to work in a noisy environment. Many recent studies have established extremely realistic weather noise generation techniques, but these methods are based on the physical characteristics of weather, and therefore require the use of

*First Author and Second Author contribute equally to this work.

979-8-3503-5409-6/24/$31.00 ©2024 IEEE
DOI 10.1109/CAI59869.2024.00156

deep information to simulate the effects of weather. However, synthetic data that are generally not realistic enough do not meet the above conditions. For this reason, we designed a simple noise generator and it has been proven effective on real severe weather data. As to the second method, we proposed to devise some way of removing noise. Doing so, it will allow the network to operate in a nearly noise-free environment. In order to achieve the purpose of noise removal, we introduced the diffusion model [25], and the characteristic of this model is that by adding noise, the network learns to remove noise. The advantage of the above approach is that it does not require the use of real-world noise for training, and this mechanism fits well with the synthetic data and noise generation methods we adopt.

In this paper, we propose the use of synthetic license plate datasets to replace the real data that are difficult to physically collect in bad weather. The biggest advantage of the above dataset is that there is no need for real data in either the foreground or the background. Our contributions are summarized below:

- Design a synthetic data generator and a method to generate severe weather noise. The proposed data generation method generates both foreground and background randomly, so no real images need to be involved except the license plate font. The YOLOv7-tiny [27] network trained using synthetic data can effectively improve the license plate recognition rate in bad weather and when day and night scenes change.
- The LD-YOLOv7 network is proposed to filter noise in YOLOv7-tiny latent space and turn it into useful information for license plate detection, thereby improving the recognition rate.
- A small license plate dataset is proposed to specifically collect license plate data in bad weather (WLP). Through this small license plate dataset, we can verify that our method can be applied to real-world data.

In the remainder of the paper, we first discuss related works, describe details of the proposed methods, present the experimental results and summarize the conclusions and future works.

II. Literature Review

In this section, we mention related works about the outdoor license plate recognition problem. In Section II-A we describe the past works about how to generate synthetic data and how to add weather noise to the synthetic data. As to how to reduce the impact of noise on license plate recognition will be reviewed in II-B. In II-C, we will review how previous researches dealt with license plate recognition system.

A. Synthetic data and noise generator

Adding noise is necessary for successful use of synthetic datasets in outdoor environment. Björklund et al. [4] used real images as the background, and then combined them with the synthetic data of the foreground. They also used data enhancement techniques, such as illumination changes, viewing angle changes, random color shifts, etc., to prove that adding noise can effectively improve the accuracy of outdoor license plate recognition. In addition, Wrenninge and Unger [29] built a synthetic dataset of street scenes using photorealistic rendering techniques. They also added camera sampling noise to the data, and the price of producing the realistic synthetic data was high.

In order to solve the extreme data problems caused by severe weather, Hahner et al. [7] proposed to use the method of [29] as a basis, compared with in-depth information to try to change the content of the dataset to foggy weather, and directly use [29]'s label to create a severe weather dataset. There are some excellent methods [22] [10] that also use depth information to simulate severe weather, and these methods have indeed resulted in better rain and fog removal effects. However, depth information is not as easy to obtain as RGB information, so it is more expensive to collect data.

In recent years, there have been some studies using CycleGAN [32] style transfer methods to simulate severe weather [20] [1] [18] [19] [16]. However, similar methods require existing datasets and labels to generate images. Because license plates are different in different areas, a license plate recognition system that has been trained and feasible in one area will not work in other places. Therefore, the style transfer method can only overcome the weather problem. As for the license plate content that change with the region, the style transfer method cannot handle it.

Based on the above analysis, we create a synthetic license plate generator based on license plate rules to replace real data. At the same time, to ensure the randomness of each license plate, we generate datasets and labels in real time during training. In order to reduce the burden on the system, we use a simple noise generator to simulate severe weather, because the style transfer noise generation method is too expensive.

B. De-noising method

Eliminating noise is an important means to improve license plate recognition rate. There have been many related studies in the past. For example, Azam et al. [2] proposed filtering to remove rain and fog in severe weather. Svoboda et al. [26] use the characteristics of CNN to learn the dynamic blur matrix of a fixed-angle camera in a real scene, and use it to invert the blurring process to achieve a clear image. In [23], Seibel et al. introduced timeline video information to find clear versions of blurred license plates at other time points in a tracking manner. In [14], Liu et al. take a similar approach that uses the ReID system to collect multiple license plates from the same vehicle, and then use generative adversarial network to combine the information of the multiple license plates to generate super-resolution license plates for identification. However, the above method needs to collect multiple time points and even images from multiple cameras to work effectively. In [11], Kerim et al. proposed to train a network with multiple branches using synthetic data on semantic segmentation tasks. In this network, each branch needs to be trained separately on semantic segmentation, weather, day and night changes, etc., and finally it

is expected that the semantic segmentation results will not be affected by noise. But the above method needs the branches to be established independently for each type of noise, which will lead to a very complex system.

Another method often used for noise removal is diffusion model [25]. It can define the original image and the noisy image as two distributions, and use Markov chain to define the process of converting from a clean image to a noisy image, and learn the distribution transformation in between. In [8], Ho et al. proposed Denoising Diffusion Probabilistic Models (DDPM) that established a paradigm for image generation from Gaussian noise. The method they proposed is basically a simplification of diffusion model. Trying to add the previous Gaussian noise to a clear image is equivalent to the result of accumulating multiple Gaussian noise in the past. In [21], Rombach et al. proposed a diffusion model that works in latent space. They tried to change the location of diffusion from the RGB space to the latent space of the large language model, which resulted in better outcome for each task. Since the noise of real data cannot obtain the corresponding noise-free version, in order to solve the problem of matching noisy data and clean data, Wei et al. [28] used cycle-consistent architecture to train the diffusion model with already matched data, and used another network to transfer the results to the real data. In [3], Bansal et al. argued that the diffusion model does not need to rely on Gaussian noise, and they thus proposed the cold-diffusion mechanism. Their method can convert a variety of noisy images back into clean images. In 2023, Luo et al. [15] proposed using synthetic data to train rain and fog removal diffusion model, and they also confirmed that this method can really remove rain and fog on real images. In [17] Özdenizci and Legenstein used conditional diffusion model to concatenate the noisy images under severe weather, such as rain, fog, and snow with Gaussian noise images. Their method can simultaneously remove Gaussian noise and noise caused by severe weather.

Based on the above methods, we believe that the diffusion model can indeed help remove severe weather noise, but it will require high training and inference costs if it handles noise in the image space [21]. This work is based on the concept of cold-diffusion [3], and mixes it with various noises on the synthetic dataset. At the same time, the noise of the image space is substituted into the latent space to improve the accuracy of license plate recognition.

C. License plate detection and recognition

Several past methods mentioned that the license plate recognition capability can be effectively improved if the system is divided into multiple steps. For example, the method proposed in [12] and [13] is to split the system into vehicle detection, license plate detection, and license plate recognition. In [24], Silva and Jung proposed to add an automatic affine transformation step into the system to reduce the impact of license plate angle differences on recognition. Thanks to the rapid advancement of object detection technology, a method [4] using only a few steps has achieved very good results. We used YOLOv7-tiny [27] as the license plate detection and recognition model. We use characters and license plates together as detection targets and use post-processing to remove the license plate without characters, and merge the characters in order to obtain the final license plate recognition result.

III. Methodology

A. Our training data: license plate generator

We design a process for generating positive license plate samples, as shown in Fig. 1. We first refer to the real license plate and use it to generate samples of the license plate frame and all text. Then several words are randomly selected and arranged to form a license plate sample that meets the real specifications. We then modify the license plate text color, license plate size and rotation angle of the sample, and then combine lines and color blocks to randomly generate a background. Finally, we paste the generated license plate sample at a random position on the background, as shown in Fig. 2. The random values of all the above steps are set to situations that may occur in real life, and each text is forced to appear at least once in different positions, with different colors and sizes, thereby increasing the text features and types learned by the model.

Fig. 1. We first generate the license plate frame (left) and text that comply with regional regulations. After randomly selecting the text (middle), we arrange it on the license plate according to realistic proportions (right).

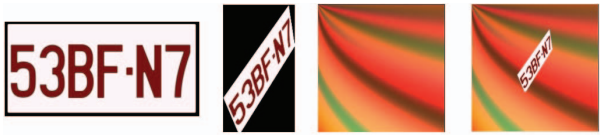

Fig. 2. Adjust the license plate text to the existing style, randomly rotate and scale it and paste it on the randomly generated background.

B. Noise generator

We observed license plate images in various actual severe weather conditions and tried to reproduce the types of noise that would cause blurred license plate text, and the results are shown in Fig. 3. In order to simulate the situation where some text is obscured due to bad weather, we randomly generate lines of different thickness to cover the license plate image. In order to simulate the difference in image brightness caused by light, we randomly select a center point in the image as the light source and linearly adjust the brightness of the image. In order to simulate text blurring caused by weather, we randomly divide the image into several blocks, and adjust all pixel values in each block to the average of all pixels in the block to achieve a mosaic-like effect. Finally, we randomly select one or more of the above noise generation methods and apply them to the synthesized license plate image.

Fig. 3. Examples of different types of noise. From left to right: original image, masking part of the text, blurring, and brightness adjustment.

In order to train the diffusion network, all noises have an intensity upper limit. We use t to represent the steps used to control the intensity of the noise. The value t from low to high represent the intensity from weak to strong. We define noise as W_t and step t as $t \in \{0, T\}, T = 100$. When $t = 0$, no noise is added, and the difference between t varies with the type of noise, as shown in Fig. 4. We assume that the intensity will increase linearly as t increases until an upper limit of intensity is reached at $t = T$.

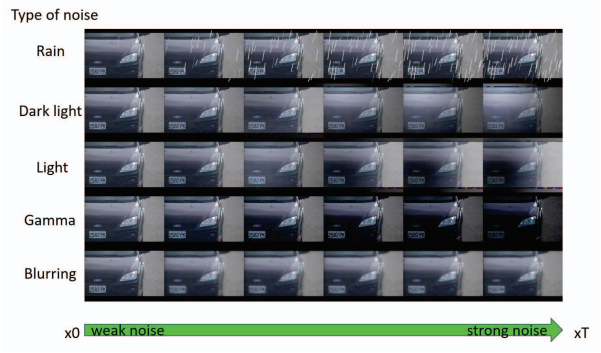

Fig. 4. Visualization of our noise generator: each column represents a different noise, with increasing intensity from left to right.

C. Training YOLOv7-tiny model

We train YOLOv7-tiny according to the preset hyperparameters of [27]. All training data are generated by III-A, and half of the data will be added with weather noise through III-B, where the noise intensity t of each data is generated with uniform distribution. Since all data are generated in units of batch size at the moment of training, our method does not have a generally defined concept of epoch. In order to facilitate the use of epoch-related optimizers, we stipulate that the number of images in an epoch is 102,400, batch size is 32, and the total number of epoch is 80. The above setting is to make the total training times approximately equal to the training times of [27] under hardware limitations.

D. Latent Diffusion model with YOLOv7-tiny(LD-YOLOv7)

The proposed latent space diffusion model actually operates in the feature space of YOLOv7-tiny, and the position is the 8x in YOLOv7, as shown in Fig. 5, and this position is also the junction of the backbone and the neck layer. We collect the features output from this location as the input of the latent diffusion model, and after removing weather noise, replace the original features with the output of the latent diffusion model and connect it to subsequent layers. We chose this location for three reasons:

1) The fewer features that change, the better.
2) This feature can affect all subsequent layers.
3) At this location we only need to change features from the same layer.

The first two points mentioned above are based on the concern of computation efficiently, and the third point is because features from different layers will have different strengths. In order to reduce the complexity of the design, we hope that the features are all provided by the same layer. Based on the above three considerations, we believe that the 8x position is the best choice.

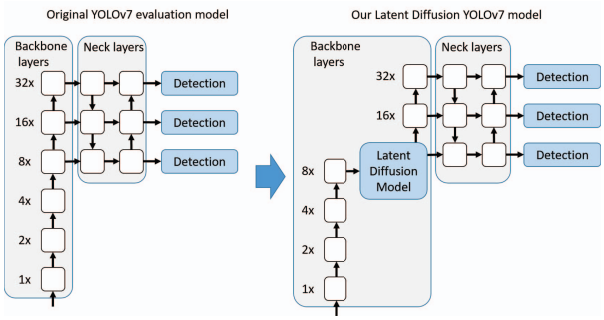

Fig. 5. Visualization of Latent Diffusion YOLOv7: diffusion model is located at the junction of backbone and neck.

The network architecture of latent diffusion model is mainly based on [3], and many improvements have been made during the training process. To maintain notational consistency, we directly utilize the notational definition of [3] as follows: Given an image $x_0 \in \mathbb{R}^N$, consider the degradation of x_0 by operator D with severity t, denoted $x_t = D(x_0, t)$. What the restoration operator R_θ does is to approximately inverse D, where $R_\theta(x_t, t) \approx x_0$, and θ represents the weight of network R.

The original cold diffusion loss is defined as follows [3]:

$$\min_{\theta} \mathbb{E}_{x \sim X} \left\| R_\theta(D(x,t),t) - x \right\|, \tag{1}$$

where x denotes a random image sampled from distribution X, and $\|*\|$ is L1 norm. Since our main objective is to remove noise rather than generate images, we modify Equation (1) to the following:

$$\min_{\theta} \mathbb{E}_{x \sim X} \left\| (D(x,t) - \epsilon_\theta(D(x,t),t)) - x \right\|, \tag{2}$$

where ϵ_θ is an ϵ network with θ weight, and the objective of Equation (1) is to directly obtain a clean image from the input image and the step t. As for Equation (2), it can be used to calculate the difference between clean images and noisy images. In order to convert the working space of the diffusion

model to the latent space, we use the layers from the top of YOLOv7 to resolution 8x as the encoder, define it as ε, and simplify $D(x,t)$ to x_t, so that the latent space of Equation (2) can be written as follows:

$$\min_{\theta} \mathbb{E}_{x\sim X} \left\| (\varepsilon(x_t) - \epsilon_\theta(\varepsilon(x_t), t)) - \varepsilon(x) \right\|, \tag{3}$$

where ϵ_θ is the latent diffusion network, and the work of cold diffusion is transferred to the latent space.

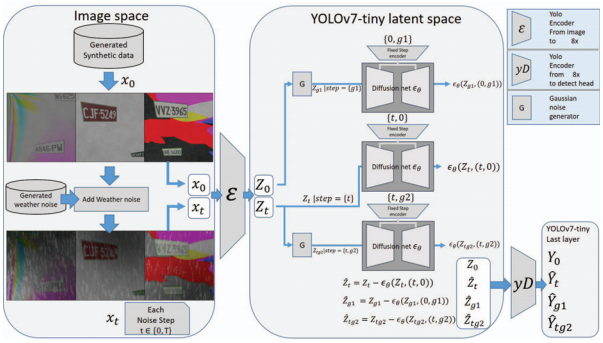

Fig. 6. Visualization of our latent diffusion training method: input images are encoded to the latent space and denoised by a latent diffusion network.

Fig. 6 shows the training method of ϵ_θ, where Generated synthetic data are described in III-A, x_0 and x_t represent the clean data and noisy data generated by the generator, Z_0 and Z_t represent the features of x_0 and x_t in the latent space. The weather noise W_t we use for training ϵ_θ is the same as III-B. In addition to weather noise, we use Gaussian noise to represent other noise in the real world that we have not considered. We generate Gaussian noise in the latent space and denote the steps of the Gaussian noise as $g \in \{0, T\}$, where $T = 100$. Under these circumstances, the Gaussian noise can be represented as $N_g(\mu, \sigma^2)$, which follows a normal distribution with the mean μ and variance σ^2, and its mean and variance are equal to the mean and variance of the input data. It should be noted that too strong Gaussian noise will have a negative impact on license plate recognition, so while maintaining the maximum T value, we assume that the value of g is uniform distribution and limit it to $T/10$. When training the ϵ_θ network, we add noise in three different manners, namely Gaussian noise only, weather noise only, and the mixture of Gaussian noise and weather noise. We use the same method as [8] [3] to do Gaussian noise sampling. Z_{g1} represents adding Gaussian noise to Z_0 at step $g1$. Since we cannot obtain the sampler that generates the weather noise in the latent space, the weather noise can only be added in the image space and then encoded to the latent space, just like Z_t mentioned earlier. Mixed noise is generated by adding Gaussian noise to Z_t in the latent space and giving an independent step $g2$. After adding Gaussian noise, Z_t is expressed as Z_{tg2}.

Based on the above symbolic definition, we can extend Equation (3) to

$$\begin{aligned} \min_{\theta} \mathbb{E}_{x\sim X} & \left\| (Z_t - \epsilon_\theta(Z_t, (t, 0))) - Z_0 \right\| \\ & + \left\| (Z_{g1} - \epsilon_\theta(Z_{g1}, (0, g1))) - Z_0 \right\| \\ & + \left\| (Z_{tg2} - \epsilon_\theta(Z_{tg2}, (t, g2))) - Z_0 \right\|. \end{aligned} \tag{4}$$

The significance of the above equation is to predict Gaussian noise, weather noise and mixed noise at the same time. After experiments, we found that the accuracy improvement of the network trained solely by relying on the above conventional diffusion loss is very limited. The reason is that the goal of Equation (4) is to predict the latent space information located at 8x in the middle of the network, but what we really care about is the output of YOLOv7. So we add the condition loss based on the detection bbox of YOLOv7, and call the encoder from 8x to the last layer of neck layer as yD, and simplify $Z_t - \epsilon_\theta(Z_t, (t, 0))$ to $\hat{Z}_t$. We simplify $\hat{Z}_{g1}$, $\hat{Z}_{tg2}$ in the same manner and encode $\{Z_0, \hat{Z}_t, \hat{Z}_{g1}, \hat{Z}_{tg2}\}$ through yD to get $\{Y_0, \hat{Y}_t, \hat{Y}_{g1}, \hat{Y}_{tg2}\}$, then we can define condition loss as follows:

$$\begin{aligned} \min_{\theta} \mathbb{E}_{x\sim X} & \left\| (\hat{Y}_t - Y_0) \right\| \\ & + \left\| (\hat{Y}_{g1} - Y_0) \right\| \\ & + \left\| (\hat{Y}_{tg2} - Y_0) \right\|. \end{aligned} \tag{5}$$

The above formula means that the noise-added data and the clean data must have the same output at the YOLOv7 detection layer after diffusion. The final loss of ϵ_θ will be Equation (4) plus Equation (5).

IV. Experiments Results

A. LD-YOLO in AOLP dataset

The license plate style and license plate dataset we generated are the same as AOLP [9], so we use this dataset to compare with other methods. AOLP is basically divided into three subsets, namely Access Control(AC), Low Enforcement(LE), Road Patrol(RP). Table I shows how the AOLP dataset is used in different ways. In Silva and Jung [24], the training was conducted by using 51 images from the LE subset of AOLP and tested only on the RP subset. In [13], Laroca et al. took 52.8% of the three subsets of AOLP as the training set, 13.2% as the validation set, and the remaining 33% as the test set. Björklund et al. [4] used all subsets for testing. In [31], Yousef et al. used two of the three subsets for training and the remaining for testing, and they repeated three times to obtain the results for the three subsets.

Since each method uses AOLP differently, in order to distinguish different settings and compare each method fairly, we combine all subsets of AOLP into one and test each method on it. The validation metric is "the degree to which the detection results of the text and numeric parts of the license plate match the ground truth." Validation results are shown in Table II. In the table, the precision and recall are calculated based on license plate detection. Plate recognition is counted

as successful only when the results completely match the ground truth label. The model labeled YOLOv7-tiny(ours) is trained from our synthetic dataset.

We test the entire AOLP using the weights and methods of [24] and [13], labeled as [24]* and [13]* in the table. Since [24] [13] is a multi-stage method, precision and recall cannot be directly compared with YOLOv7-tiny, so we focus on comparing the final license plate recognition rate. In addition, in order to eliminate the inference from the detector, we replaced the detector in [24] and [13] with YOLOv7-tiny, and retained the respective text recognition modules, marked as YOLOv7-tiny+ [24]* and YOLOv7-tiny+ [13]*. The recognition rate has been greatly improved for both methods (74.16% to 81.37% [24], 89.97% to 95.05% [13]).

It should be noted that we only use synthetic data. Although our method is not the best, the results are very close to the methods trained with real data, and even surpass some methods trained with real data. Furthermore, our proposed LD-YOLOv7 achieves a higher license plate recognition rate (from 85.12% to 87.38%) than YOLOv7-tiny. The accuracy rate is significantly increased compared with other image-based diffusion methods.

TABLE I
EXPERIMENTS WITH AOLP DATASET RELATED WORK

Models	*Plate recognition%*	*How the AOLP data is used*
[24]	RP: 98.36	train from LE test to RP
[13]	Total(AC/LE/RP): 99.2	52.8% train, 13.2%val, 33%test
[4]	AC/LE/RP: 94.6/97.8/96.9	100% test
[31]	AC/LE/RP: 97.6/97.6/94.5	2 of 3 subsets for training and the remaining for testing

TABLE II
EXPERIMENTS WITH AOLP DATASET WITH SAME METRIC

Models	*Precision%*	*Recall%*	*Plate recognition%*
[24]*	-	-	74.16
YOLOv7-tiny(ours)+ [24]*	90.96	98.52	81.37
[13]*	-	-	89.97
YOLOv7-tiny(ours)+ [13]*	90.96	98.52	95.05
YOLOv7-tiny(ours)	90.96	98.52	85.12
YOLOv7-tiny(ours)+ [3]	90.06	98.34	85.35
Our LD-YOLOv7	81.14	98.98	87.38

*Reproduce using official weights.

B. LD-YOLO in Weather License Plate dataset

In this study, we propose a Weather License Plate (WLP) dataset that contains 92 images with 116 manually annotated license plates. We use 47 images and 54 license plates as the test dataset, and the other license plates as the verification dataset. Some samples of the dataset are shown in Fig. 7. The verification set is composed of images captured from YouTube driving recorder videos, and the test set is composed of images captured from traffic surveillance camera. We use existing methods with codes to test, results are shown in Table III. We observe that models which performed well in AOLP only have 1/5 accuracy left on the WLP dataset (from 98.36% to 20.37% [24], from 99.2% to 12.96% [13]). In comparison, our LD-YOLOv7 obviously obtained better results (from 87.38% to 35.19%), which means that the proposed method is less susceptible to the noise caused by bad weather.

Fig. 7. Weather license plate dataset: All data contain at least one severe weather condition and cover day and night variations.

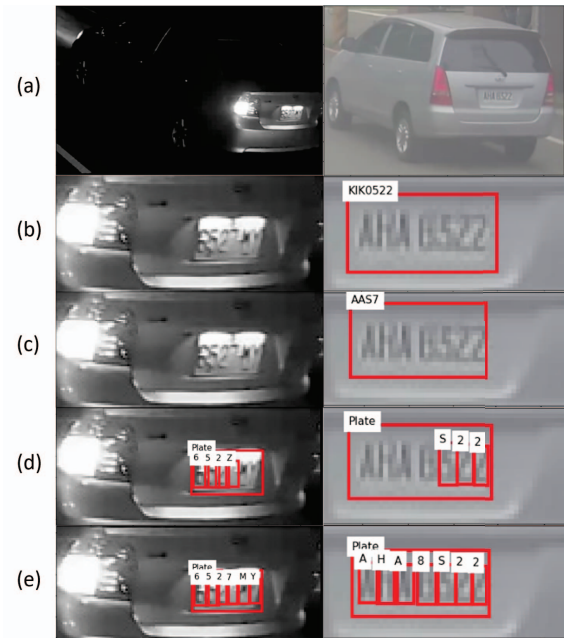

Fig. 8. Examples of detection results. Rows from top to bottom: (a) Original input images. (b) [24]. (c) [13]. (d) YOLOv7-tiny. (e) LD-YOLOv7. The images of detection results are cropped for the convenience of viewing.

Fig. 8 shows the detection results of our method on the test set. The license plate numbers in image on the left and the right columns are '6527MY' and 'AHA0322' respectively. We crop the area of license plate in the image for the convenience of viewing. The images from top to bottom are the original input images, the detection results of [24], [13], YOLOv7-tiny, and proposed LD-YOLOv7. As we can see, [24], [13] and YOLOv7-tiny may fail to detect license plate or fail to recognize plate number. The proposed LD-YOLOv7 can succeed in both plate detection and plate number recognition in image on the left, and get the lowest error rate in image on the right.

TABLE III
EXPERIMENTS WITH WEATHER DATASET

Models	*Precision%*	*Recall%*	*Plate recognition%*
[24]*	-	-	20.37
YOLOv7-tiny(ours)+ [24]*	95.35	75.93	27.77
[13]*	-	-	12.96
YOLOv7-tiny(ours)+ [13]*	95.35	75.93	31.48
YOLOv7-tiny(ours)	95.35	75.93	31.48
YOLOv7-tiny(ours)+ [3]	94.12	59.26	29.63
Our LD-YOLOv7	75.86	81.48	35.19

[a]Reproduce using official weights.

TABLE IV
ABLATION STUDY OF DIFFUSION MODELS FOR TWO DATASETS

Diffusion in which space	w/ R	w/ G	*Predict image or predict noise*	*AOLP Plate recognition%*	*WLP recognition%*
w/o diffusion				57.81	24.07
image(≈ [3])			image	56.70	25.93
image	✓		image	57.95	27.78
latent(≈ [3])			image	23.43	1.85
latent	✓		image	55.18	14.81
latent	✓	✓	image	55.13	24.07
latent	✓	✓	noise	57.39	25.93
w/o diffusion (large data)				85.12	31.48
image(large data) (≈ [3])			image	85.35	29.63
image(large data)	✓		image	85.17	33.33
latent(large data)	✓	✓	noise	87.38	35.19

V. ABLATION STUDIES

We test the model with three different diffusion settings, w/o diffusion, image diffusion, and latent diffusion, to analyze how them affect the performance of model. Details and results are shown in Table IV.

We used about 60,000 images to train YOLOv7-tiny as w/o diffusion baseline model. We trained image diffusion model and latent diffusion model with different input settings. The one marked with "≈ [3]." is using the same settings with [3]. w/ R and w/ G represent reconstruction loss and Gaussian noise explained in Section III-D. The details of "Predict image or predict noise" also described in Section III-D. All testings were conducted on the AOLP and WLP datasets. We mark the results that perform below w/o diffusion in red, and the results that perform above w/o diffusion in green. As we can see in table IV, latent diffusion method ≈ [3] has a great negative impact on the system (dropped from 57% to 23% in AOLP, and from 24% to 1% in WLP). The proposed improvements, adding reconstruction loss and Gaussian noise, both have a positive impact on latent space diffusion. Those improvements even greatly improves the performance of license plate recognition when testing WLP dataset (upgraded from 1% to 14% after introducing reconstruction loss in WLP, and from 14% to 24% after introducing Gaussian noise in WLP).

The lower part of Table IV shows the verification results of training diffusion models with large amount of data. This is for proving whether we can improve model performance by training with purely generated data in large quantities. The only difference between the models listed above and below in table IV is the amount of training data. We can see that these heavily generated synthetic data improve all methods, and the latent diffusion approach outperforms the image diffusion approach. We speculate that these models can achieve a better performance is because the latent diffusion approach relies heavily on high-quality features, training with large amount of synthetic data help the latent space obtaining appropriate features more effectively.

TABLE V
COMPARE THE IMPACT OF CLASSES FOLLOWING REAL/UNIFORM DISTRIBUTION

Distribution of	*AP% A to Z*	*AR% A to Z*	*AP% 0 to 9*	*AR% 0 to 9*	*AOLP Plate recognition%*
Real	38.26	18.54	46.27	33.15	64.42
Uniform	35.41	32.05	68.68	32.29	85.12

Additionally, the significantly higher likelihood of numeric categories appearing on license plates compared to English letters creates an imbalance that negatively impacts the system's performance. To support our claim, we employed a synthetic data generator to create two datasets for training the yolov7-tiny network. The first dataset mimics the imbalanced distribution observed in real license plates, whereas the second dataset features a uniform distribution, in which the occurrence probabilities of numbers and English letters are identical. As demonstrated in Table V, the yolov7-tiny network trained with uniformly distributed data demonstrates superior overall performance on the AOLP dataset compared to the network trained with a real data distribution. This is despite having slightly lower average precision (AP) for English letters and average recall (AR) for numbers than the network trained on the real data distribution. Such a result confirms that synthetic data can effectively address the issue of category imbalance in real license plate data.

In addition, to analyze the impact of various simulated noises on the model, we used a noise generator to produce different types of noise, and trained the YOLOv7-tiny network separately for each type. Overall, as shown in Table VI, compared to scenarios where no specific noise was introduced, the model's overall performance on both the AOLP and WLP datasets improved after introducing specific noises through the noise generator. More specifically, in the WLP dataset, training with all types of noise integrated can significantly enhance accuracy; whereas in the AOLP dataset, introducing rain and blurring noise has the most substantial impact on performance improvement. This confirms that to enable the model to effectively handle adverse weather conditions, simulating a combination of all types of noise is essential. Conversely, for AOLP data collected under normal weather conditions, there is no need to introduce excessive simulated noise.

VI. CONCLUSION

In this paper, we demonstrate the potential of purely synthetic license plates paired with deep learning methods. Most

TABLE VI
THE IMPACT OF EACH NOISES

Models	*AOLP Plate Recognition%*	*WLP Plate Recognition%*
YOLOv7-tiny w/o noise generator	59.98	14.81
w/ rain noise only	74.91	20.37
w/ light noise only	69.82	16.67
w/ gamma noise only	68.90	18.52
w/ blurring noise only	71.49	20.37
w/ all noise	69.55	24.07

of the previous research relied on real license plate dataset, and the recognition results of our proposed YOLOv7-tiny on the Taiwanese license plate dataset AOLP even exceeded some models trained using real data. This means that it is feasible to use purely synthetic data to replace real license plate data that is difficult to collect. We also proposed LD-YOLOv7, which is a method that brings the image space noise into the feature space and removes noise. This method can further improve the license plate recognition results on AOLP to 87.38%. At the same time, we verified the effectiveness of this method on real data on the proposed severe weather dataset (WLP), and the results also show that our method can actually withstand the impact of severe weather.

REFERENCES

[1] Asha Anoosheh, Torsten Sattler, Radu Timofte, Marc Pollefeys, and Luc Van Gool. Night-to-day image translation for retrieval-based localization. arXiv, 2019.

[2] Samiul Azam and Md Monirul Islam. Automatic license plate detection in hazardous condition. *Journal of Visual Communication and Image Representation*, 36:172–186, 2016.

[3] Arpit Bansal, Eitan Borgnia, Hong-Min Chu, Jie S. Li, Hamid Kazemi, Furong Huang, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Cold diffusion: Inverting arbitrary image transforms without noise. arXiv, 2022.

[4] Tomas Björklund, Attilio Fiandrotti, Mauro Annarumma, Gianluca Francini, and Enrico Magli. Robust license plate recognition using neural networks trained on synthetic images. *Pattern Recognition*, 93:134–146, 2019.

[5] Shan Du, Mahmoud Ibrahim, Mohamed Shehata, and Wael Badawy. Automatic license plate recognition (alpr): A state-of-the-art review. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(2):311–325, 2013.

[6] Yanxiang Gong, Linjie Deng, Shuai Tao, Xinchen Lu, Peicheng Wu, Zhiwei Xie, Zheng Ma, and Mei Xie. Unified chinese license plate detection and recognition with high efficiency. arXiv, 2022.

[7] Martin Hahner, Dengxin Dai, Christos Sakaridis, Jan-Nico Zaech, and Luc Van Gool. Semantic understanding of foggy scenes with purely synthetic data. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, Oct. 2019.

[8] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[9] Gee-Sern Hsu, Jiun-Chang Chen, and Yu-Zu Chung. Application-oriented license plate recognition. *IEEE Transactions on Vehicular Technology*, 62(2):552–561, 2013.

[10] Xiaowei Hu, Chi-Wing Fu, Lei Zhu, and Pheng-Ann Heng. Depth-attentional features for single-image rain removal. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[11] Abdulrahman Kerim, Felipe Chamone, Washington Ramos, Leandro Soriano Marcolino, Erickson R. Nascimento, and Richard Jiang. Semantic segmentation under adverse conditions: A weather and nighttime-aware synthetic data-based approach. arXiv, 2022.

[12] 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 *International Joint Conference on Neural Networks (IJCNN)*, pages 1–10, July 2018.

[13] Rayson Laroca, Luiz A. Zanlorensi, Gabriel R. Gonçalves, Eduardo Todt, William Robson Schwartz, and David Menotti. An efficient and layout-independent automatic license plate recognition system based on the yolo detector. *IET Intelligent Transport Systems*, 15(4):483–503, Feb. 2021.

[14] Wu Liu, Xinchen Liu, Huadomg Ma, and Peng Cheng. Beyond human-level license plate super-resolution with progressive vehicle search and domain priori gan. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1618–1626, 2017.

[15] Ziwei Luo, Fredrik K Gustafsson, Zheng Zhao, Jens Sjölund, and Thomas B Schön. Refusion: Enabling large-size realistic image restoration with latent-space diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 1680–1691, 2023.

[16] Valentina Muşat, Ivan Fursa, Paul Newman, Fabio Cuzzolin, and Andrew Bradley. Multi-weather city: Adverse weather stacking for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pages 2906–2915, October 2021.

[17] Ozan Özdenizci and Robert Legenstein. Restoring vision in adverse weather conditions with patch-based denoising diffusion models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–12, 2023.

[18] Horia Porav, Tom Bruls, and Paul Newman. Don't worry about the weather: Unsupervised condition-dependent domain adaptation. arXiv, 2019.

[19] Horia Porav, Tom Bruls, and Paul Newman. I can see clearly now : Image restoration via de-raining. arXiv, 2019.

[20] Horia Porav, Will Maddern, and Paul Newman. Adversarial training for adverse conditions: Robust metric localisation using appearance transfer. arXiv, 2018.

[21] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. arXiv, 2022.

[22] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Semantic foggy scene understanding with synthetic data. *International Journal of Computer Vision*, 126(9):973–992, Mar. 2018.

[23] Hilário Seibel, Siome Goldenstein, and Anderson Rocha. Eyes on the target: Super-resolution and license-plate recognition in low-quality surveillance videos. *IEEE access*, 5:20020–20035, 2017.

[24] S. M. Silva and C. R. Jung. License plate detection and recognition in unconstrained scenarios. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 580–596, Sep 2018.

[25] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2256–2265, Lille, France, 07–09 Jul 2015. PMLR.

[26] Pavel Svoboda, Michal Hradiš, Lukáš Maršík, and Pavel Zemcík. Cnn for license plate motion deblurring. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3832–3836. IEEE, 2016.

[27] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[28] Mingqiang Wei, Yiyang Shen, Yongzhen Wang, Haoran Xie, Jing Qin, and Fu Lee Wang. Raindiffusion: When unsupervised learning meets diffusion models for real-world image deraining. arXiv, 2023.

[29] Magnus Wrenninge and Jonas Unger. Synscapes: A photorealistic synthetic dataset for street scene parsing. arXiv, 2018.

[30] Zhenbo Xu, Wei Yang, Ajin Meng, Nanxue Lu, and Huan Huang. Towards end-to-end license plate detection and recognition: A large dataset and baseline. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 255–271, 2018.

[31] Mohamed Yousef, Khaled F. Hussain, and Usama S. Mohammed. Accurate, data-efficient, unconstrained text recognition with convolutional neural networks. *Pattern Recognition*, 108:107482, 2020.

[32] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. arXiv, 2020.