2024 6th International Conference on Communications, Information System and Computer Engineering (CISCE) | 979-8-3503-5317-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/CISCE62493.2024.10653221

Traffic Surveillance License Plate Recognition System in Heavy Rainfall

Guoci Cai, Lihan Tong, Jie Pan, Ming Chen, Zhengxuan Huang, Yiliang Wu *
College of Ocean Information Engineering
Jimei University
Xiamen, China
* Corresponding author: wuyiliang@jmu.edu.cn

***Abstract*—In adverse weather conditions such as heavy rainfall, visibility obstruction significantly reduces the image quality captured by surveillance cameras, impacting the recognition of vehicles and their license plates. This leads to ineffective monitoring of traffic violations and other situations. To address this issue, we have developed a vehicle and license plate detection and recognition system specifically designed for heavy rain conditions, which consists of three main components: (a) image de-raining, (b) license plate localization, and (c) license plate information extraction. For image de-raining, we have introduced the NeRD-Rain algorithm and created a de-raining dataset containing vehicles by adding random noise. We then utilized pre-trained models for transfer learning. In the license plate localization stage, we employed the YOLOv5 model for accurate localization of license plates. Lastly, for license plate information extraction, we constructed a character extraction network. Through experiments conducted on our dataset, the complete version and lightweight version of the de-raining module achieved PSNR scores of 38.42 and 36.82, respectively. Moreover, the accuracy of license plate information extraction reached 96.33%.**

Keywords- Image de-raining; Vehicle detection; License plate recognition; Transfer learning

I. INTRODUCTION

In adverse weather conditions, traffic surveillance systems often face severe challenges, especially during heavy rainfall. In such circumstances, rain, fog, and blurred visibility significantly degrade the image quality captured by surveillance cameras, thereby affecting the effective monitoring and identification of passing vehicles. This not only increases the workload of traffic management departments but also poses certain risks to urban security management.

Under adverse weather conditions, criminals may exploit the weather's effects to evade surveillance, thereby increasing the difficulty of detecting criminal activities. By enhancing the license plate recognition capabilities of traffic surveillance systems in adverse weather, it is possible to strengthen the monitoring and combating of criminal behavior, improve urban security levels, and maintain social stability. Through research on license plate recognition technology under adverse weather conditions, it is possible to effectively enhance the system's capabilities to cope with adverse weather, thereby improving the accuracy and stability of license plate recognition.

In recent years, image enhancement, particularly in areas such as rain removal, has become a major focus. With the widespread development of deep learning, the use of deep learning models has become the mainstream approach to solving many image problems. Methods based on deep learning [1]–[3] have surpassed early traditional algorithms and demonstrated excellent restoration performance. Recently, researchers have attempted to replace CNNs with Transformers as the basic structure for visual tasks [4]. Driven by the significant success of the visual Transformer in modeling non-local information, frameworks based on Transformers [5], [6] have emerged to better remove rain. Chen et al. designed an effective multiscale Transformer [7], which utilizes multiscale rainfall information to generate high-quality rain removal results.

In object recognition, both accuracy and detection speed are key concerns for researchers. Over the past decade, more and more research teams have focused on model compression, model lightweighting, or using single-stage methods for detection tasks. The YOLO series [8]–[11], as a major player in this field, exhibit strong performance in both detection accuracy and speed, and are widely applied in engineering fields with high demands on detection efficiency.

In the field of text recognition, methods based on deep learning neural networks also play a significant role. With the continuous development and improvement of deep learning technology, these neural network methods have become the mainstream choice for text recognition research and applications[12], [13]. They play a crucial role in improving text recognition accuracy, robustness, and generalization ability. Compared to traditional feature engineering-based methods, deep learning neural network methods have stronger universality and adaptability, enabling them to handle text recognition tasks under various fonts, sizes, styles, and complex backgrounds.

II. METHOD

A. General Route

Figure1 illustrates the overall architecture of the network. It takes an image $I \in \mathbb{R}^{C\times H\times W}$ as input, which then passes through three networks: rain removal, license plate detection and localization, and license plate information recognition. The outputs are the rain-removed image, the license plate localized image, and the license plate recognition result, respectively.

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DE GOIAS. Downloaded on September 25,2024 at 23:01:45 UTC from IEEE Xplore. Restrictions apply.

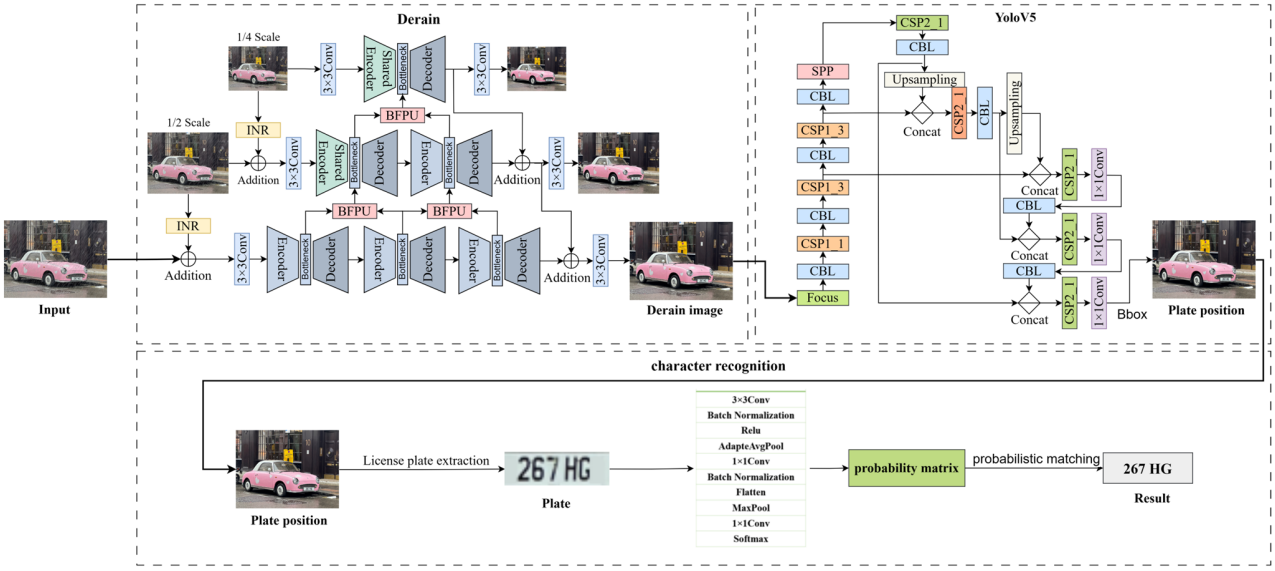

Figure1. Overview of the network. The network consists of three main components: the rain removal module processes the image input to remove rainwater; subsequently, the YOLOv5 detection network detects the license plate; finally, the detected area is passed to the character recognition network for license plate recognition, thereby extracting the license plate information.

B. *Image De-raining*

The NeRD-Rain team developed an effective bidirectional multiscale Transformer with implicit neural representations, called NeRD-Rain, for image rain removal. It aims to better explore multiscale information and simulate complex rain patterns. NeRD-Rain includes an intrascale INR branch and an interscale bidirectional branch BFPU. The former learns latent degradation representations from different rain-in images, while the latter facilitates richer collaborative representations across different scales.

To jointly learn traditional representations based on UNet and continuous representations based on INR in a multiscale manner, the network undergoes end-to-end training using a hybrid loss function. The Charbonnier loss L_{char} [14], frequency loss L_{freq}[15], and edge loss L_{edge}[16] are employed to constrain the learning of specific scales. Additionally, the $L1$ norm is utilized to prevent color shifts during INR prediction of RGB. The overall loss related to INR, based on a coarse feature grid and a fine feature grid, is calculated as follows:

$$L_{inr} = \sum_{s=1}^{2} \|I_s - T_s\|_1 \quad (1)$$

Where Is and Ts represent the s-scale reconstructed image and the s-scale target ground truth image for INR, respectively. The proposed loss function L_{total} for network training is defined as:

$$L_{total} = L_{char} + \alpha_1 L_{freq} + \alpha_2 L_{edge} + \alpha_3 L_{inr} \quad (2)$$

The scalar weights, $\alpha 1$, $\alpha 2$, and $\alpha 3$ are respectively set to 0.01, 0.05, and 0.1.

C. *License Plate Segmentation and Information Extraction*

In this system, we employ the YOLOv5-S network for the object detection task to locate license plates. YOLO (You Only Look Once) is a popular object detection algorithm known for its speed and efficiency. Compared to traditional object detection algorithms, YOLO treats object detection as a regression problem, predicting bounding boxes and classes for all objects in an image through a single forward pass network.

The main modules are illustrated in Figure2.

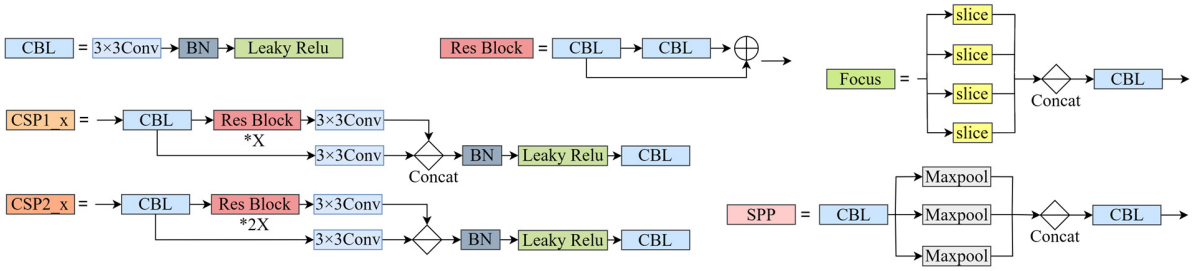

Figure2. Main Modules of YOLOv5

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DE GOIAS. Downloaded on September 25,2024 at 23:01:45 UTC from IEEE Xplore. Restrictions apply.

In YOLOv5, the CIOU (Complete IoU) loss [15] is employed as the loss function for the target bounding box. CIOU is a loss function that combines IoU (Intersection over Union) and GIoU (Generalized IoU) for bounding box regression tasks in object detection. Its formula is as follows:

$$L_{CIOU} = 1 - \left(IoU - \frac{c^2}{a^2} - \frac{p^2}{a^2} + \lambda v \right) \tag{3}$$

Afterward, the recognition of license plate characters is conducted. To achieve better recognition, the characters on the license plate are first segmented. To distinguish between the possible upper and lower layers of the license plate, the plate is horizontally divided into 5/12 and 1/3 portions. Subsequently, each portion is resized to the original size and then concatenated horizontally. For this process, assuming the input is $I_y \in \mathbb{R}^{H\times W\times 3}$, the output becomes $I_p \in \mathbb{R}^{\frac{2}{3}H\times 2W\times 3}$.

A license plate character library is predefined, including possible characters, special vehicle identifiers, digits, letters, etc. After the model outputs corresponding probabilities, the character with the highest probability is selected for character matching, resulting in the corresponding license plate result.

For this part of the network, the input is the segmented license plate obtained from the previous YOLO network output, denoted as $I_p \in \mathbb{R}^{\frac{2}{3}H\times 2W\times 3}$. The output consists of a series of probabilities $P_i \in \mathbb{R}^{78}$, where i represents the i-th character. After matching probabilities, the character corresponding to the highest probability in the character library is the target character.

In character recognition, cross entropy is used as the loss function, and its calculation formula is as follows:

$$L_{ce}(y,\hat{y}) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\log\left(\widehat{y_{ij}}\right) \tag{4}$$

Among them, N is the number of samples, M is the number of categories, y_{ij} is the true probability that the i-th sample belongs to the j-th category,$\widehat{y_{ij}}$ is the probability that the i-th sample predicted by the model belongs to the j-th class.

III. EXPERIMENT

A. Simple rain removal data set containing vehicles

Due to the absence of rainfall data containing vehicles and for the sake of dataset supervision, many mainstream rainfall datasets currently adopt the method of artificially synthesizing rain traces to create datasets [17]–[20]. Inspired by this, we created a simple rainfall dataset containing vehicles.

We recorded three videos (1080p 30fps) and split them into frames. We then artificially introduced random noise to "simulate rainfall" and construct the dataset. Among these, two one-minute segments were used as the training set, while one forty-second segment was divided into two parts and used as the test set and validation set, respectively (with a ratio of 6:2:2). Before and after processing, the average PSNR of corresponding images in the dataset was 25.74.

Figure3 illustrates an overview of the rainfall simulation process.

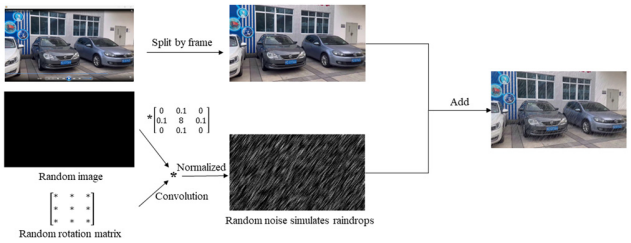

Figure3. Overview of the Data Image Processing Process

On one hand, the video is split into frames to obtain image a. On the other hand, a random rain input image of the same size as a is generated. Random noise is added to this image, and then a random rotation matrix is used to perform Gaussian blur on the rotated diagonal matrix, producing a blurred raindrop effect. Convolving this effect with the random image simulates raindrops, which are then superimposed onto image a, resulting in a simulated rainy day image.

B. Train

Training was conducted using the Nvidia RTX 4090 (24GB RAM), an 18-core AMD EPYC 9754 128-Core Processor, PyTorch 1.11.0, and CUDA 11.3. The training lasted for 3000 epochs. The Adam optimizer was utilized with an initial learning rate of 1e-4, gradually decreasing to a final rate of 1e-6. Two versions of the model were trained: the full version with a transformer hidden layer depth of 48, and the lightweight version with a transformer hidden layer depth of 32.

Performance evaluation was conducted using Peak Signal-to-Noise Ratio (PSNR), a metric commonly used to measure the quality of images or signals. It is often employed to assess the effectiveness of compression algorithms or the level of distortion in image restoration processes.

The calculation formula is as follows:

$$PSNR = 10 \cdot log_{10}\left(\frac{MAX^2}{MSE}\right) = 20\, log_{10}\left(\frac{MAX}{\sqrt{MSE}}\right) \tag{5}$$

Among them, MAX represents the maximum possible amplitude of the signal, and MSE represents the mean square error. The calculation formula is:

$$MSE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}[I(i,j) - K(i,j)]^2 \tag{6}$$

TABLE I. shows the performance results of the post-training test.

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DE GOIAS. Downloaded on September 25,2024 at 23:01:45 UTC from IEEE Xplore. Restrictions apply.

TABLE I. TRAINING SET PERFORMANCE COMPARISON RESULTS

type	*Param*	*Model*	*PSNR1*	*PSNR2*
Pre-trained model	10530314	Rain200L-S[20]	26.68	30.62
	100314	Rain200H-S[20]	26.35	30.38
	10530314	DDN-S[17]	29.23	29.6
	10530314	DID-S[19]	30.13	30.99
	10530314	SPA-S[18]	25.97	30.26
	22893146	Rain200L-L[20]	27.59	30.61
Transfer learning	10530314	Self-S	36.83	34.77
	22893146	Self-L	**38.42**	**35.16**

The results on PSNR1 are test results using the pre-trained model on our homemade dataset. PSNR2 is an image that has never appeared in the training set/test set/validation set. Compared with using the pre-trained model directly, its value increases significantly after transfer training. Compared with the provided pre-trained model, it is more suitable for actual engineering needs.

Through data and after transfer learning, its performance has been greatly improved. Even the lightweight version is enough to meet the needs of engineering. Figure4 shows the visual comparison effect. It can also be seen intuitively through visualization that after transfer learning, the effect is significant, with basically no traces of rain visible, and it can better meet engineering needs.

Figure4. Visualization and effect comparison chart

C. License Plate Segmentation and Information Extraction

Input the image after the rain has been removed, input it into the yolov5 network for license plate detection, and get the marking result. Use the detected license plate position information to perform affine transformation and input it into the character recognition network to get the result.

To measure the identification accuracy in the statistical test, three experiments were conducted, and the results are shown in TABLE II.

TABLE II. LICENSE PLATE RECOGNITION ACCURACY RESULTS

Experimental rounds	Accuracy
1	95%
2	98%
3	96%

After testing, its average accuracy rate reached more than 96.33%, which has a high accuracy rate and can better complete the recognition task.

IV. CONCLUSIONS

In this study, we primarily focus on the license plate recognition technology under adverse weather conditions. To address the challenge of rain in images, we employ the NeRD-Rain algorithm for image deraining and construct a simple deraining dataset. We conduct transfer learning on pre-trained models using our dataset. On our data, the full version and lightweight version of the model achieve PSNR values of 38.42 and 36.82, respectively, representing approximately a 38% improvement over the pre-trained model. Moreover, visually, the results exhibit significant enhancement, with rain traces being virtually indistinguishable. Following the image deraining process, we utilize the YOLOv5 object detection network for license plate detection and localization. Additionally, we define a character library for license plates and employ a custom-built character extraction network to detect the probability of each character in the library, thereby obtaining specific characters. In the future, our work will focus on image deraining under real rainy conditions to contribute to image deraining and traffic monitoring systems in real weather scenarios.

REFERENCES

[1] Y. Ba *et al.*, "Not Just Streaks: Towards Ground Truth for Single Image Deraining," *European Conference on Computer Vision. Cham: Springer Nature Switzerland*, pp. 723–740, Jun. 2022.

[2] X. Chen *et al.*, "Unpaired Deep Image Deraining Using Dual Contrastive Learning," *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2017--2026, Jan. 2022.

[3] X. Chen, J. Pan, J. Dong, and J. Tang, "Towards Unified Deep Image Deraining: A Survey and A New Benchmark," *arxiv:2310.03535*, Oct. 2023.

[4] A. Dosovitskiy *et al.*, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," *ICLR*, Jan. 2021, doi: arXiv.2010.11929.

[5] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, and M.-H. Yang, "Restormer: Efficient Transformer for High-Resolution Image Restoration," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2022. doi: 10.1109/cvpr52688.2022.00564.

[6] H. Chen *et al.*, "Pre-Trained Image Processing Transformer," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2021. doi: 10.1109/cvpr46437.2021.01212.

[7] X. Chen, J. Pan, and J. Dong, "Bidirectional Multi-Scale Implicit Neural Representations for Image Deraining," *arXiv.2404.01547*.

[8] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement.," *arXiv: Computer Vision and Pattern Recognition*, Apr. 2018.

[9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016. doi: 10.1109/cvpr.2016.91.

[10] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO Series in 2021," Jul. 2021.

[11] A. Bochkovskiy, C.-Y. Wang, and H.-Y. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," *Cornell University - arXiv*, Apr. 2020.

[12] C. Yao, "DocXChain: A Powerful Open-Source Toolchain for Document Parsing and Beyond," *arxiv:2310*, vol. 12430, Oct. 2023.

[13] Y. Shi *et al.*, "EXPLORING OCR CAPABILITIES OF GPT-4V(ISION) : A QUANTITATIVE AND IN-DEPTH EVALUATION," *arxiv:2310*, vol. 16809, Jan. 2023.

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DE GOIAS. Downloaded on September 25,2024 at 23:01:45 UTC from IEEE Xplore. Restrictions apply.

[14] S. W. Zamir *et al.*, "Multi-Stage Progressive Image Restoration," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2021. doi: 10.1109/cvpr46437.2021.01458.

[15] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019. doi: 10.1109/cvpr.2019.00075.

[16] K. Jiang *et al.*, "Multi-Scale Progressive Fusion Network for Single Image Deraining," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020. doi: 10.1109/cvpr42600.2020.00837.

[17] H. Zhang and V. M. Patel, "Density-aware Single Image De-raining using a Multi-stream Dense Network," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2018. doi: 10.1109/cvpr.2018.00079.

[18] T. Wang, X. Yang, K. Xu, S. Chen, Q. Zhang, and R. W. H. Lau, "Spatial Attentive Single-Image Deraining With a High Quality Real Rain Dataset," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019. doi: 10.1109/cvpr.2019.01255.

[19] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, and J. Paisley, "Removing Rain from Single Images via a Deep Detail Network," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017. doi: 10.1109/cvpr.2017.186.

[20] W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan, "Deep Joint Rain Detection and Removal from a Single Image," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017. doi: 10.1109/cvpr.2017.183.

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DE GOIAS. Downloaded on September 25,2024 at 23:01:45 UTC from IEEE Xplore. Restrictions apply.