A Federated Learning-based License Plate Recognition Scheme for 5G-enabled Internet of Vehicles

Xiangjie Kong, *Senior Member, IEEE,* Kailai Wang, Mingliang Hou, Xinyu Hao, Guojiang Shen, Xin Chen, and Feng Xia, *Senior Member, IEEE,*

***Abstract*—License plate is an essential characteristic to identify vehicles for the traffic management, and thus license plate recognition is important for Internet of Vehicles. Since 5G has been widely covered, mobile devices are utilized to assist the traffic management, which is a significant part of Industry 4.0. However, there have always been privacy risks due to centralized training of models. Also, the trained model cannot be directly deployed on the mobile device due to its large number of parameters. In this paper, we propose a federated learning-based license plate recognition framework (FedLPR) to solve these problems. We design detection and recognition model to apply in the mobile device. In terms of user privacy, data in individuals is harnessed on their mobile devices instead of the server to train models based on federated learning. Extensive experiments demonstrate that FedLPR has high accuracy and acceptable communication cost while preserving user privacy.**

***Index Terms*—Intelligent Transportation System, federated learning, license plate recognition, transfer learning, 5G, Internet of Vehicles.**

I. Introduction

INTELLIGENT Transportation System (ITS) is one of the important measures to create an intelligent/digital city. The development of ITS is beneficial to improve public safety and travel of residents [1], which accelerates Industry 4.0 [2]. In terms of Internet of Vehicles (IoV), license plate information is used as the unique identifier on roads and public transportation. However, according to the official statistics, the number of vehicles (2020) in China has reached 360 million totally, which brings new challenges and problems to the management and control of vehicles in the transportation system, such as privacy issues, and large resource consumption in large cities. Therefore, it is of great research significance and practical value to study the automatic license plate recognition (LPR) in IoV to improve service of management by detecting and recognizing vehicle license plates on roads [3].

This work was supported in part by the National Natural Science Foundation of China under Grant 62072409 and Grant 62073295, in part by the Zhejiang Provincial Natural Science Foundation under Grant LR21F020003, and in part by the Fundamental Research Funds for the Provincial Universities of Zhejiang under Grant RF-B2020001. *(Corresponding authors: Feng Xia)*

X. Kong, and G. Shen are with the College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China (e-mail: xjkong@ieee.org; gjshen1975@zjut.edu.cn).

K. Wang, M. Hou, X. Hao, and X. Chen are with the School of Software, Dalian University of Technology, Dalian 116620, China (e-mail: kailai.w@qq.com; teemohold@outlook.com; Hao.XinYu@outlook.com; chenxin20@mail.dlut.edu.cn).

F. Xia is with School of Engineering, IT and Physical Sciences, Federation University Australia, Ballarat 3353, Australia (e-mail: f.xia@ieee.org)

As far as an actual phenomenon is concerned, a lot of vehicle owners park their vehicles at will for their own convenience, which is illegal behavior. As a result of the large number of such behaviors, it further disrupts the order of urban transportation, and the personal safety of residents is seriously threatened while traveling. Though many cameras are mounted at road junctions to help supervise and inspect illegal parking, it is still difficult to recognize covered license plates of vehicles due to complicated traffic situations. Therefore, traffic police are required to control these illegal parking behaviors. For example, they use mobile devices to take photos of illegal parking vehicle and give owners tickets. Based on the requirement of recognizing license plate with the mobile device under 5G, it is significant to research how to use the mobile terminals and artificial intelligence to detect and recognize the license plate information immediately.

Reviewing methods of license plate detection (LPD), these methods can be divided into two categories. On the one hand, traditional visual feature methods usually utilize regular shape [4] and specific pixel color [5] of license plate. On the other hand, deep learning can effectively extract features to detect objects [6] and recognize objects [7], that is, deep learning-based methods are also able to detect license plates. However, owing to a large number of parameters, they are incapable to be directly deployed on mobile devices, which leads to more frequent connections with the server and increases communication overhead. Therefore, it is important that an edge computing-based model [8] is used to reduce communication cost.

As for license plate identification, common methods divide it into two sub-tasks: character segmentation and character recognition, but the methods are easily affected by blurring, noise, and deformation of character directions. The other methods usually use Convolutional Neural Networks (CNN) to identify the entire license plate and directly outputs the classification code. In terms of user privacy, most LPR methods based on deep learning are applied with centralized training. This method requires training data to be centralized in a certain server or a single data center for manual annotation, which is time-consuming. Furthermore, there is a risk of license plate information leakage in the process of data centralization. In recent years, user privacy has become a crucial problem. China and some European countries have strengthened the preservation of user data through different regulations [9], [10]. Therefore, how to train the model has become a critical issue in practice while restricting the collection of user data and preserving user privacy. Besides, another important issue is data isolated islands. As shown in Fig. 1, since different companies have different data centers, when users use different applications of different companies, their data will naturally be stored in different data centers and cannot be exchanged. As a result, it is difficult to use these valuable data at the same time to train powerful models. Moreover, the current artificial intelligence market is dominated by tech giants who offer cloud-based artificial intelligence solutions and APIs that make it impossible for users to control data, while these companies can monopolize data through data centralization. Therefore, users are passive to their information data and privacy cannot be guaranteed.

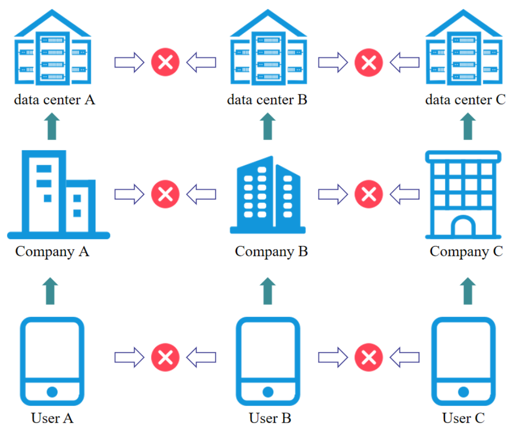


Fig. 1: The data islanding problems in industrial application.

In this paper, in order to solve the above problems, we proposed a federated learning-based framework for license plate recognition (FedLPR). Firstly, based on the edge computing architecture, we design the LPD model and significantly reduce the number of parameters, which makes it possible to deploy directly at the mobile devices. Then, we present a simple but effective LPR model and utilize a tilt license plate correction algorithm to improve the accuracy of license plate character recognition. Based on models, we combine federated learning to preserve user privacy, which is a decentralized optimization process, that is, it uses user individual data at the model level. Under this framework, users do not need to upload data to a central server, instead use limited computing resources of the mobile device to train the local model by individual data based on the global model from the central server. The update gradients of parameters of local models are uploaded to the server, which would be used to update the parameters of the global model. Therefore, the old global model will be replaced with a more powerful model based on data from all users. In addition, transfer learning is used to optimize the local model. Finally, we implement a mobile application instance, which aims at the management of illegal parking vehicles in the practical various scene through LPD and LPR under the age of 5G.

The main contributions are summarized as follows:

- We propose a novel federated learning-based framework for plate recognition (FedLPR) with the aim of making use of computing resources of the mobile device to train model with low latency and consumption, and keep pace with preserving user privacy.
- We design a LPD and LPR model, which can be deploy on mobile devices and ensure the accuracy and efficiency of detection. Considering the fact that the direction of shooting is often inclined, we present a tilt license plate correction algorithm to improve the accuracy of character recognition of license plate.
- We construct extensive experiments to evaluate our model. The results demonstrate that our detection and recognition models are outstanding in both accuracy and speed, especially in complicated actual environments. Finally, we implement a mobile application to test FedLPR and further verify its efficiency.

The rest of this paper is organized as follows. In Section II, we review the related work about LPD, LPR, and federated learning. In Section III, we introduce our proposed FedLPR and model of in detail. We present experiment settings and analysis of result In Section IV. Following that, we demonstrate test results of our realized mobile terminal system based on our method in Section V. Finally, we summarize and provide further discussion in Section VI.

II. Related Work

A. License Plate Detection

At present, the LPD systems mainly adopt artificially defined image features or deep learning for intelligent detection.

1) Method based on artificially defined visual features: Regular side of rectangle of license plates in the image can be used to determine the existence of license plates, such as combining edge statistics with seed growth strategy or mathematical morphology [4]. Besides, the license plate contains characters composing a string, that is, structural features are obvious. Therefore, the license plate can be positioned by detecting the characters in the image. For example, Li et al. [11] used MSER algorithm to extract the license plate character region. In addition, Pixel color is the basic feature of license plates, which is also used for LPD. Asif et al. [5] proposed a cascade license plate classifier based on color saliency features to identify the real license plates in candidate regions. Because of the regular pixel texture distribution of the license plate area, texture features are also applied to the license plate location algorithm [12].

2) Method based on deep learning technology: The prior LPD methods based on visual features has hit a developmental bottleneck due to various factors, such as the complicated environment, and unstable illumination in the images. Deep learning has outstanding performance in object detection [6], [13], [14]. The multi-target detection algorithm based on this has greatly improved compared with previous algorithms. Therefore, a lot of LPD methods are proposed based on deep learning. Xie et al. [15] proposed the MD-YOLO model, which is a multi-directional LPD framework based on CNN. Differently, our method focuses on the application on mobile devices, which can make full use of computing resource and improve the efficiency of LPD.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2021.3067324, IEEE Transactions on Industrial Informatics

B. License Plate Recognition

There are usually two stages in LPR, including character segmentation and character recognition. For character segmentation, common approach includes projection algorithm [16], SIFT feature [17], and extreme region extraction [18]. However, these methods have the problems of segmentation deviation, because they are easily affected by blurring, noise, and deformation of character directions in the images, which directly lead to recognition errors. Besides, some other methods are proposed to directly recognize characters of license plate without character segmentation. For example, Zherzdev et al. [19] proposed a lightweight LPR network, LPRNet, which can directly output the recognition results with high accuracy and real-time processing speed.

C. Federated Learning

In 2017, Google introduced Federated Learning [20], which is originally designed to train machine learning models with the data distributed acorss multiple mobile devices, and achieves the purpose of preserving user privacy at the same time. Since then, federated Learning has aroused widespread interest among researchers, which focus on data privacy, communication efficiency, and statistical heterogeneity [21]. For example, Zhang et al. [22] present a platform architecture of blockchain-based federated learning systems to tackle the challenge of data heterogeneity in failure detection of intelligent interconnection of things. Ma et al. [4] analyze potential security issues of wireless end-user equipment in Federated Learning through simulation experiments, and provide possible solutions. Besides, Hu et al. [23] proposed a novel inference framework, federated regional learning (FRL), to improve computational efficiency for urban environment sensing, which combines edge computing and distributed deep learning.

III. Design of Framework

In this section, we illustrate the details of procedures and models in our proposed FedLPR.

A. Overview

FedLPR aims to realize effective LPD and recognition by federated learning and assist traffic police to manage illegal vehicles without compromising privacy safety. As shown in Fig. 2, first of all, the initial models of LPD and LPR are trained based on real dataset by the server. Then, the global model on the server is distributed to all users as their local model on mobile devices. In the process of using their local model, personal data would be collected, that is annotated images after LPD. The mobile device would utilize limited computing resource to further train its local model based on personal data. Considering the differences in hardware performance between the server and the mobile device, transfer learning is used to improve model training. After training, the FederatedAveraging Algorithm is employed to update the global model through uploading local models of users, where personal data is not shared and parameters of model would be encrypted. Finally, users will receive an updated model, which can recognize the license plate characters more accurately. It is noteworthy that the server will not directly contact the user data, which can effectively preserve user privacy.

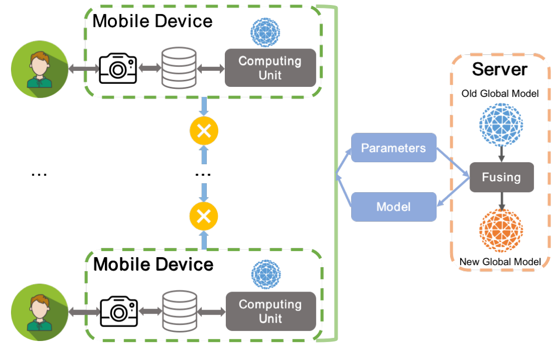


Fig. 2: The framework of FedLPR.

B. License Plate Detection

Considering the actual scene, both accuracy and efficiency of detection are required. Inspired by YOLO [13] and Mask-RCNN [24], we design a fast LPD model, which can be applied to the mobile devices. As shown in Fig. 3, the whole network architecture consists of three modules: fast feature map module, preliminary detection module, and mask branch module.

In fast feature map module, feature map of high resolution images need to be rapidly extracted to accelerate downsampling. Besides, we focus on keeping the number of parameters as few as possible, which enables the model to be deployed on mobile devices. Therefore, we employ depthwise separable convolutional block (DSCB) [25] instead of standard convolutional operation. DSCB separates standard convolution the two steps of traditional convolution, that are depthwise and pointwise convolution. The parameter calculation amount in DSCB and the parameter calculation amount in standard convolutional (SC) operation are as follows:

$$\begin{cases} \mathrm{C}_{DSCB} = D_F \times C_{in} \times k^2 + D_F \times C_{in} \times C_{out}, \\ \mathrm{C}_{SC} = D_F \times k^2 \times C_{in} \times C_{out} \end{cases} \tag{1}$$

where C_{DSCB} and C_{SC} are the parameter calculation amount in DSCB and SC, respectively, D_F is the size of feature map, k is the kernel size of convolution, C_{in} and C_{out} are respectively the number of input and output channels. DSCB reduces by $1/C_{out} + 1/k^2$ compared with SC.

Preliminary detection module is responsible for initial detection of license plates. Similarly, depthwise separable convolutional blocks are used to extract feature map. In our case, the size of the feature map is 40×40, which is enough to detect license plate. Moreover, the number of object classification, c, is 1 (e.g. the license plate). Then, following the idea of YOLO, the input image is divided into an $n \times n$ grid, where each grid cell would compute the confidence score that reflects the probability of a license plate in its corresponding m bounding boxes (BBox). The BBox can be represented by (x, y, w, h, Co), where (x, y) is the center coordinate of BBox,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2021.3067324, IEEE Transactions on Industrial Informatics

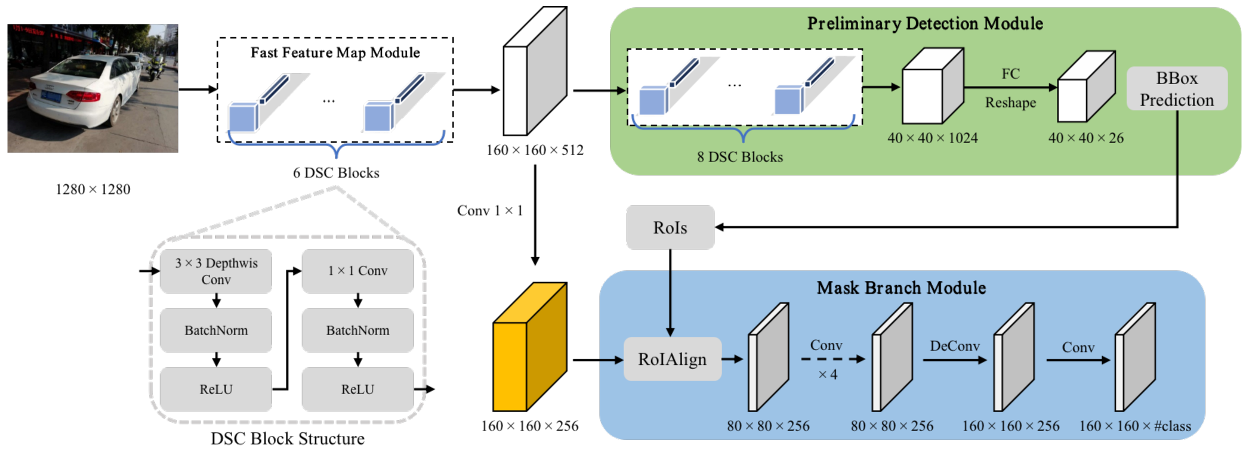


Fig. 3: The architecture of LPD model consisting of fast feature map, preliminary detection, and mask branch modules.

(w, h) is the size normalization of its width and height relative to the raw image, and Co denotes confidence scores (that is the intersection over union (IOU) between the predicted box and the ground truth). Therefore, after the final fully connected layer, the shape of the detection convolutional kernel is $1\times1\times26$ (e.g. $1\times1\times(m\times5+c)$, where $m=5$, and $c=1$ in our model). Besides, we use parallel non-maximal suppression (NMS) to accelerate the fixing of predicted BBox.

The shape of license plate is regular quadrangle, which means adjacent pixels may belong to the same license plate instance, and mask is continuous in spatial layout. Therefore, we add a mask branch module to complete the task of license plate segmentation (pixel-to-pixel). In mask branch module, RoIAlign [24] is firstly adopted to align feature map with region of interest (RoI), which is obtained from predicted BBox in previous preliminary detection module. After several convolutional and DeConvolutional operations, we compute a mask prediction matrix, M, by sigmoid function as the result of a binary classification.

We define the loss function, which includes classification loss ($\mathcal{L}_{cls}$), BBox loss ($\mathcal{L}_{box}$), confidence score loss ($\mathcal{L}_{con}$), and mask loss ($\mathcal{L}_{mask}$). The equation is as follows:

$$\begin{cases}\mathcal{L}=\lambda_{cls}\mathcal{L}_{cls}+\lambda_{box}\mathcal{L}_{box}+\lambda_{con}\mathcal{L}_{con}+\lambda_{mask}\mathcal{L}_{mask},\\ \mathcal{L}_{cls}=BCE(P,\hat{P}),\\ \mathcal{L}_{box}=\sum_{i=0}^{n^2}\theta_i^*[(x_i-\hat{x}_i)^2+(y_i-\hat{y}_i)^2+\\ \quad(\sqrt{w_i}-\sqrt{\hat{x}_i})^2+(\sqrt{h_i}-\sqrt{\hat{h}_i})^2],\\ \mathcal{L}_{con}=\sum_{i=0}^{n^2}\theta_i^*(Co_i-\hat{Co}_i)^2,\\ \mathcal{L}_{mask}=BCE(M,\hat{M})\end{cases} \tag{2}$$

where, λ represents hyperparameter, P is the probability matrix of classification, θ_i^* is an indicator function, that is, $\theta_i^*=1$ if the license plate appears in cell i, or else $\theta_i^*=0$, M is mask prediction matrix, and BCE is binary cross-entropy function.

C. License Plate Recognition

Based on the result of detection, the license plate image is cropped to recognize characters. Empirically, the segmentation result of license plates has a certain tilt in both horizontal and vertical directions. Therefore, it is essential to correct tilt license plates to improve accuracy of recognition. We are able to correct horizontal tilt by angle calculation and rotation operation. In our case, we focus on vertical tilt. First of all, we binarize the image covered by the mask after horizontal tilt correction, which helps subsequent tilt judgment. Then, we select specific pixel rows that are located at 1/4, 2/4, and 3/4 of the height of the binary image in our case, and calculate the length of continuous $pixel=0$ starting from the first pixel value to judge tilt level. Obviously, the white area is detected as a parallelogram if the length of these three rows is relatively largely different, such as an increasing or decreasing trend. In the correction algorithm, we utilize affine transformation to correct the image detected as a parallelogram, including slope calculation, coordinates calculation of key points, and coordinate transformation. The formula of slope calculation can be written as:

$$\kappa=\min(|\frac{len_2-len_1}{H_{LP}}\times2-\tan(angle)|,\\ |\frac{len_3-len_1}{H_{LP}}\times2-\tan(angle)|) \tag{3}$$

where len_1, len_2, and len_3 are the length continuous $pixel=0$ corresponding to three specific rows, H_{LP} and W_{LP} denotes the height and width of the license plate image. Therefore, three original key point (upper left, upper right, and lower left) can be respectively represented as $(|\kappa|H_{LP},0)$, $(W_{LP}-1,0)$, and $(0,H_{LP}-1)$, which are transformed as $(|\kappa|H_{LP}|/2,0)$, $(W_{LP}-1-|\kappa|H_{LP}|/2,0)$, and $(|\kappa|H_{LP}|/2,H_{LP}-1)$ in the target image.

After correction, Chinese license plate consists of Chinese character, letters, and numbers. We use Support Vector Machine (SVM) classifier to distinguish Chinese characters and non-Chinese characters. Considering limited computing resource of the mobile device, we need to segment characters through binary image, gradient sharpening, and threshold denoise, which are recognized one by one. In our case, the image size of segmented each character is reshaped as 36×28. We design a end-to-end CNN as show in Fig. 4. We train two models to recognize Chinese characters and non-Chinese characters, respectively, whose different shape of output results from specific Chinese character in the license plate belonging to one of 31 Chinese character while non-Chinese characters composed of 24 letters and 10 numbers. In particular, these models can be retrain by the mobile devices.

D. Federated Learning

Our FedLPR adopts federated learning to preserve user privacy and realize the relearning and retraining of LPR model on the mobile device. We assume that our given data is from N different users, which can be represented as $D_1, D_2, \cdots, D_N$. These data is the images of vehicles that are ticketed by the traffic police. In conventional centralized training, the model, $\mathcal{M}_{all}$, is trained on central serer by integrating all the data, $D = D_1 \sqcup D_2 \sqcup \cdots \sqcup D_N$. However, in our case, a model based on federated learning, $\mathcal{M}_{Fed}$, is train on the mobile device, where user i need not transfer its data D_i to central server or each other. Meanwhile, we make sure that the accuracy of $\mathcal{M}_{Fed}$ is close or superior to $\mathcal{M}_{all}$. In our FedLPR, the loss function of each user i can be defined as follows:

$$\mathcal{L}(D_i) = \frac{1}{2|D_i|} \sum_{(x_j, y_j) \in D_i} \boldsymbol{f}(\mathcal{M}^i_{Fed}(x_j), y_j), \tag{4}$$

where (x_j, y_j) denotes a data sample, and $\boldsymbol{f}(\cdot)$ is a objective function like Mean Square Error (MSE) function.

During the process of training, the weight gradients of user local model parameter set, $\nabla\mathcal{W} = \nabla\omega_1, \nabla\omega_2, \cdots, \nabla\omega_N$, would be calculated and transfered to the central server. Inspired by FederatedAveraging [20], we design weighted parameters update algorithm, that is, we would evaluate the efficiency of each gradient with the test set on the server to assign weights for parameter updates of the global model. The formula to update global model parameters is denoted as follows:

$$W_{new} = W_{old} - \eta \sum_{\nabla\omega_i \in \nabla\mathcal{W}} \frac{F1_i}{F1_{old}} \nabla\omega_i, \tag{5}$$

where η is the learning rate, and $F1$ denote the evaluation index (F1-score) of models based on the same test set. Therefore, the new global model is updated based on all user data and possesses the ability of generalization.

E. Transfer Learning

For different users, there is individual characteristic information in their own local model due to significant difference among user data, which is caused by various factors such as time (day or evening), and weather (rain or shine) of given images. In our FedLPR, we introduce transfer learning to optimize the performance of local model. In the recognition model, we froze the weights of convolutional layers and downsampling layers which are used to extract low-level features of license plate character. The parameters of fully connected layers would be updated during training, which tends to learning individual characteristic. Therefore, the objective function of the local model can be written by:

$$\mathcal{O}^i = \mathcal{L}(D_i) + \frac{1}{d^2}\boldsymbol{f}(C_s, C_i), \tag{6}$$

where $C_s, C_i \in \mathbb{R}^{d \times d}$ is the covariance matrix of d-dimensional feature from the server and user i, respectively.

TABLE I: The statistics and shape of the dataset during the training

Name	#	Shape
Photos	32,100	1280×960
License plates	53,000	128×50
Users	2,000	-
Characters	-	36×28

IV. EXPERIMENTS

In this section, we will introduce the details of experimental dataset and settings. Then, we show the evaluation and analysis of results.

A. Dataset and Training Details

Our dataset includes 32,100 high-quality photos (1280×960), 60% of which are from the traffic police in Fujian, Chain. After artificial statistics, there are approximately 53,000 images of license plates in these photos. On the one hand, for the LPD model, the dateset is divided into two sets that are 70% training set, and 30% test set. On the other hand, in order to simulate the application scene of federated learning, we randomly selected 2,000 detected license plate images as the training set, which is used to train the initial global model of LPR on the server. Then, we assume that there are 2,000 users in the application. Similarly, the remaining of license plate images is divided into 70% training set, and 30% test set, which are assigned to these users. Each independent user would possess a random number of images ($|D_i| \in [100, 1000]$) as the user personal set to train the local model. The basic information and shape of experimental data during training are shown in TABLE I. Besides, we show examples of original photos in Fig. 5.

In the LPD model, $\lambda_{cls}, \lambda_{box}, \lambda_{con}$, and, λ_{mask} are set to 1, 1.5, 1.5, and 6.125. The training of LPD model is divided into two stages since the mask branch module is highly depended on the BBbox prediction from backbone network, namely, fast feature map module and preliminary detection module. In the first stage, we only train the backbone network. After 400 epochs, the average precision (AP) reaches about 0.83. Then, we froze the weight of backbone network, and continue to train the mask branch module based on the RoIs.

For the simulation experiment of federated learning, the standard LPR model is trained with centralized method, which is able to achieve 95.3% accuracy after 1000 iterations. In our FedLPR, considering users online status and computing resources, we choose 15% of users for each parameter update. In the process of training of local models, we use Adam optimizer with 0.01 learning rate.

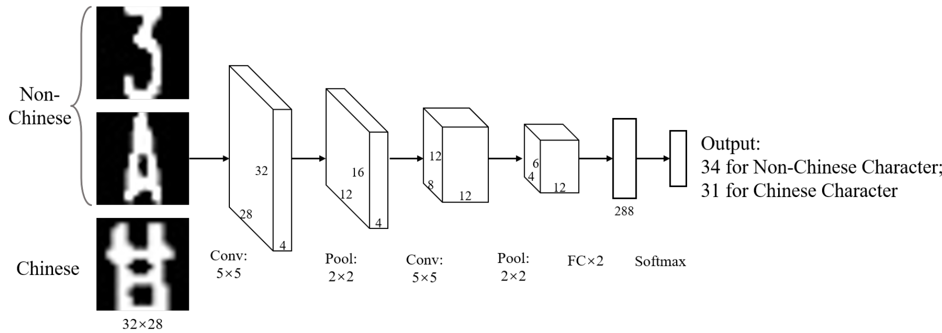


Fig. 4: The architecture of LPR model based on CNN.

Fig. 5: The examples of original photos.

TABLE II: The result of LPD

Method	Precison (%)	Recall (%)	FPS
Mask-RCNN [24]	95.3	97.2	9.6
YOLO [26]	94.3	85.4	33.6
[27]	93.2	86.7	14.5
[28]	95.1	92.5	19.0
[29]	95.2	97.3	17.6
Ours (standard NMS)	**96.7**	**97.8**	29.0
Ours (parallel NMS)	94.3	95.7	**36.6**

In addition, our detection model is implemented based on the Keras and TensorFlow framework on the server with NVIDIA RTX 2080 Ti. The experimental mobile environment is the Android phone with Qualcomm Snapdragon 855 processor to train the local model of LPR.

TABLE III: The result of LPR

Method	Accuracy (%)	Speed (ms)
[27]	88	**0.24**
[18]	77	0.33
[30]	82	0.38
Ours	**94**	0.36

TABLE IV: The comparison result of federated learning algorithm

Method	Proportion of Dirty Data		
	0.2%	2%	5%
FederatedAverage [20]	93	81	70
Ours (Weighted)	**94**	**90**	**83**

TABLE V: The comparison result on the mobile device

Characters	General Scene		Illegal Parking	
	Ours	HyperLPR	Ours	HyperLPR
All	97	90	94	65
Non-Chineses	**100**	**94**	**97**	**80**

B. Results

1) License Plate detection: For the LPD, our method is evaluated against other methods [24], [26]–[29] based on our dataset in TABLE II. Our detector is superior to others both in accuracy and rate of detection. In terms of accuracy, the performance of our model is similar to Mask-RCNN, and is even slightly better. Benefiting by DSCB, the speed of our model is greatly improved while ensuring the accuracy of detection compared with others. Besides, we design the simple ablation experiment to illustrate the improvement of detection speed. Based on the evaluation result with standard NMS and with parallel NMS, we prove that our model is further improved in speed, and the decrease in accuracy caused by the parallel NMS is acceptable.

Fig. 6: The examples of license plates after tilt correction.

2) License Plate Recognition with Federated Learning: In this part, we firstly apply the tilt license plate correction algorithm in previous detection result. The examples of cropped license plates after tilt correction are shown in Fig. 6. As show in Fig. 6, we can recognize the characters in the license plates more easily from the perspective of human vision, which can prove the efficiency of tilt correction algorithm. Compared

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2021.3067324, IEEE Transactions on Industrial Informatics

(a) General environment

(b) Large-angle tilt environment

(c) Too strong or dark light environment

(d) Long distance environmente

Fig. 7: The examples of test data of different environments.

with other LPR methods [18], [27], [30], our evaluation result has the best performance in accuracy based on 100 photos of illegal parking as shown in TABLE III. We find that single horizontal tilt correction in other methods cannot solve the problem of character deformation in the picture of illegal parking vehicles, which leads to the low accuracy of recognition. In terms of speed, our model is not the best due to the addition of the tilt correction algorithm, while it is sufficient to ensure real-time performance in actual scenes. Moreover, we consider the impact of dirty data on the model in the process of federated learning, and our weighted parameter update algorithm is compared with FederatedAverage as show in TABLE IV. From the result, our method is more robust.

Furthermore, in order to further prove the efficiency on the mobile device, we compare our model with HyperLPR[1]. According to the content of photos, we test methods in the two scenes separately: general scene and illegal parking scene. For the general scene, there are no tilt license plates in the picture. By contrast, the photos are taken of vehicles parking illegally, where the license plates are usually tilted. As shown in TABLE V, the result of our model is similar to HyperLPR in the general scene. However, for the illegal parking scene, 3% reduction of the accuracy occurs in our model, while 25% sharp reduction in HyperLPR.

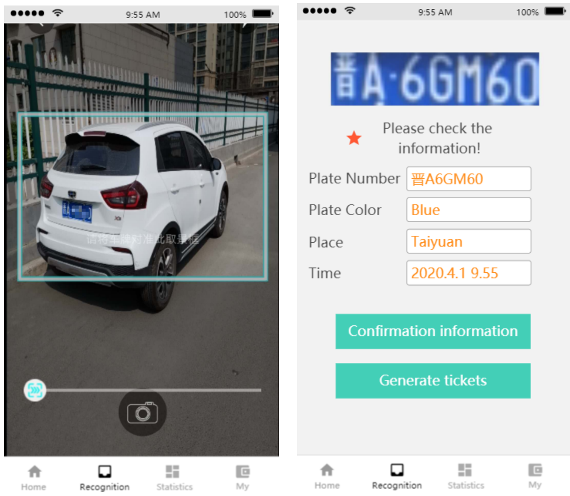
[1]A open source Chinese license plate recognition framework, which can be applied in the Android phone: https://github.com/zeusees/HyperLPR

V. LICENSE PLATE RECOGNITION MOBILE APPLICATION

In this section, we present a case study, a mobile LPR APP, for the practical scene of illegal parking vehicles as show in Fig. 8.

Firstly, we utilize Keras to train our LPD model, which would be transformed to TensorFlow format. Therefore, the relevant Java code can be written to perform the inference of TensorFlow model in our Android application after adding dependency item (e.g. TensorFlow Mobile). Besides, LPR model is completed through DeepLearning4J. Then, we deploy our FedLPR on the mobile device including LPD and LPR models. Considering performance, MySQL is utilized to implement the database of the APP. The APP obtains the photos through calling the camera of the mobile phone. Then, the license plate result would be displayed to users, which is detected and recognized by models without data transmission. Besides, the collected data and results are used to retrain the local recognition model on the mobile device, where the update gradient of parameters is calculated and transmitted to the server. Finally, the server would fuse all gradients to update the global model, which will be distributed to all users.

Meanwhile, considering communication cost, each user usually generate about communication cost of 80MB. The server need receive the transmission parameters of 300 users ($2000\times15\%$) to update the global model for each round, which add up to 23.4GB during the whole process of optimization.



(a) Photo acquisition page (b) Result display page

Fig. 8: The demo of license plate recognition APP.

TABLE VI: The comparison result on the mobile device

Environment	Recognition (%)	Detection (%)
(1)	94	100
(2)	90	100
(3)	64	74
(4)	78	90

Under the speed of 5G network, the communication cost is acceptable for our model update.

In order to verify the efficiency of our APP in various situations, we carried out a series of simulation experiments, including general environment (1), large-angle tilt environment (2), too strong or dark light environment (3), and long distance environment (4) as shown in Fig. 7. For each situation, we test 100 collected photos. From TABLE VI, environment (1), (2), and (4) have high accurate result of LPD, while the LPR accuracy of environment (4) is relatively low. For environment (3), the result of LPD and LPR is relatively poor due to the reflected light of license plate characters. In summary, our FedLPR can be effectively applied in the mobile device for edge computing, which can meet the basic requirement of precision and speed for LPD and LPR while ensuring the user privacy.

VI. Conclusion

In this article, we proposed a federated learning-based LPR framework (FedLPR) to preserve user privacy for IoV. To our knowledge, it is the first time that federated learning has been applied in license plate recognition on the mobile device. We design LPD and LPR models that can be deployed on the mobile device and ensure both accuracy and speed. Besides, tilt license plate correction algorithm is used to adapt the practical scene to improve the efficiency of license plate recognition. In our FedLPR, the training of recognition model is executed on the mobile device, what needs to be transmitted to the server is only the gradients of parameters instead of user data itself. Therefore, we solve the problem of user privacy. Although our method has outperformed some other methods, detection and recognition models need further improvement and discussion for complicated and diverse actual scenes, such as accuracy, acceleration, and multi-characters recognition. In addition, the mobile APP also has high expansion value to IoV in the future.

References

[1] L. Zhu, F. R. Yu, Y. Wang, B. Ning, and T. Tang, “Big data analytics in intelligent transportation systems: A survey,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 1, pp. 383–398, 2019.

[2] X. Zhou, Y. Hu, W. Liang, J. Ma, and Q. Jin, “Variational lstm enhanced anomaly detection for industrial big data,” *IEEE Transactions on Industrial Informatics*, 2020.

[3] T. Andreica and B. Groza, “Secure v2v communication with identity-based cryptography from license plate recognition,” in *2019 Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS)*, 2019, pp. 366–373.

[4] M. Vargas, S. L. Toral, F. Barrero, and F. Cortés, “A license plate extraction algorithm based on edge statistics and region growing,” vol. 5716, pp. 317–326, 2009.

[5] M. R. Asif, Q. Chun, S. Hussain, and M. S. Fareed, “Multiple licence plate detection for chinese vehicles in dense traffic scenarios,” *IET Intelligent Transport Systems*, vol. 10, no. 8, pp. 535–544, 2016.

[6] Z. Zhao, P. Zheng, S. Xu, and X. Wu, “Object detection with deep learning: A review,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3212–3232, 2019.

[7] X. Zhou, W. Liang, K. I. Wang, H. Wang, L. T. Yang, and Q. Jin, “Deep-learning-enhanced human activity recognition for internet of healthcare things,” *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6429–6438, 2020.

[8] X. Kong, S. Tong, H. Gao, G. Shen, K. Wang, M. Collotta, I. You, and S. Das, “Mobile edge cooperation optimization for wearable internet of things: A network representation-based framework,” *IEEE Transactions on Industrial Informatics*, 2020.

[9] N. Inkster, *China’s Cyber Power*. Routledge, 2018.

[10] M. Goddard, “The eu general data protection regulation (gdpr): European regulation that has a global impact,” *International Journal of Market Research*, vol. 59, no. 6, pp. 703–705, 2017.

[11] B. Li, B. Tian, Y. Li, and D. Wen, “Component-based license plate detection using conditional random field model,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 4, pp. 1690–1699, 2013.

[12] M. Wafy and A. M. Madbouly, “Efficient method for vehicle license plate identification based on learning a morphological feature,” *IET Intelligent Transport Systems*, vol. 10, no. 6, pp. 389–395, 2016.

[13] 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)*, June 2016, pp. 779–788.

[14] X. Kong, K. Wang, S. Wang, X. Wang, X. Jiang, Y. Guo, G. Shen, X. Chen, and Q. Ni, “Real-time mask identification for covid-19: An edge computing-based deep learning framework,” *IEEE Internet of Things Journal*, 2021.

[15] L. Xie, T. Ahmad, L. Jin, Y. Liu, and S. Zhang, “A new cnn-based method for multi-directional car license plate detection,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 2, pp. 507–517, 2018.

[16] Y. Wen, Y. Lu, J. Yan, Z. Zhou, K. M. von Deneen, and P. Shi, “An algorithm for license plate recognition applied to intelligent transportation system,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 830–845, 2011.

[17] Y. Wang, X. Ban, J. Chen, B. Hu, and X. Yang, “License plate recognition based on sift feature,” *Optik*, vol. 126, no. 21, pp. 2895–2901, 2015.

[18] 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*, vol. 17, no. 4, pp. 1096–1107, 2016.

[19] S. Zherzdev and A. Gruzdev, “Lprnet: License plate recognition via deep neural networks,” *arXiv:1806.10447*, 2018.

[20] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, vol. 54. PMLR, 2017, pp. 1273–1282.

[21] S. K. Lo, Q. Lu, C. Wang, H. Paik, and L. Zhu, "A systematic literature review on federated machine learning: From a software engineering perspective," *arXiv:2007.11354*, 2020.

[22] W. Zhang, Q. Lu, Q. Yu, Z. Li, Y. Liu, S. K. Lo, S. Chen, X. Xu, and L. Zhu, "Blockchain-based federated learning for device failure detection in industrial iot," *IEEE Internet of Things Journal*, 2020.

[23] B. Hu, Y. Gao, L. Liu, and H. Ma, "Federated region-learning: An edge computing based framework for urban environment sensing," in *2018 IEEE Global Communications Conference (GLOBECOM)*, 2018, pp. 1–7.

[24] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[25] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv:1704.04861*, 2017.

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

[27] H. Li, P. Wang, M. You, and C. Shen, "Reading car license plates using deep neural networks," *Image and Vision Computing*, vol. 72, pp. 14–23, 2018.

[28] S. Montazzolli and C. Jung, "Real-time brazilian license plate detection and recognition using deep convolutional neural networks," in *2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, 2017, pp. 55–62.

[29] M. A. Rafique, W. Pedrycz, and M. Jeon, "Vehicle license plate detection using region-based convolutional neural networks," *Soft Computing*, vol. 22, no. 19, pp. 6429–6440, 2018.

[30] 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)*, September 2018.