A Hybrid Deep Learning Algorithm for the License Plate Detection and Recognition in Vehicle-to-Vehicle Communications

Xiuqin Pan, Sumin Li, Ruixiang Li, and Na Sun

***Abstract*— With the rapid development of Internet of Things (IoT) in the field of transportation, the vehicle-to-vehicle (V2V) communication not only becomes available on a large scale, but also will be an indispensable part of the future transportation. License plates are the identification of vehicles, so the license plate detection and recognition in the V2V communication scenario is very important. However, the existing license plate detection and recognition methods are suffering from a low accuracy rate issue. To solve this issue, we propose a hybrid deep learning algorithm as the license plate detection and recognition model by fusing YOLOV3 and CRNN. The proposed model enables the network itself to better utilize the different fine-grained features in the high and low layers to carry out multi-scale detection and recognition. In this model, we utilize the fast and accurate performance of YOLOV3, and the excellent detection ability of CRNN. As a result, this proposed model reaps the benefit of both. Finally, we test this proposed model in difficult scenarios and low-quality license plate images caused by weather, and results show this proposed license plate detection and recognition model can achieve a higher mean average precision, better comprehensive performance, and excellent robustness.**

***Index Terms*— Hybrid deep learning, vehicle-to-vehicle communications, license plate detection and recognition, YOLOV3, CRNN.**

I. Introduction

VEHICLE-TO-VEHICLE (V2V) communications refer to the wireless exchange of various information between vehicles. V2V communications allow vehicles to send and receive omni-directional information, thus creating a 360-degree perception in other vehicles. V2V is the application of Internet of Things (IoT) in the field of Intelligent Transportation System (ITS). The continuous development of IoT enables increasingly more complex application of V2V. Vehicles can communicate faster with more information, and then make more intelligent decisions. V2V can provide drivers with information that is beyond human's 300-meter limit of line of sight, which can be further impaired by weather, traffic conditions or terrain. At the same time, IoT allows vehicles to communicate with not only other vehicles but also surrounding environment as well. More specifically, it can communicate with infrastructures such as the road, sensors, etc., which is called vehicle-to-infrastructure (V2I). Also, vehicles can communicate with pedestrians with mobile devices as well, which can help vehicles understand their immediate surroundings better, which is usually the key factor in understanding whether an accident is likely to occur. This type of communication with pedestrians is call vehicle-to-pedestrian (V2P). V2V utilizes radar and camera sensors equipped on vehicles to collect information, and visualization is one of the most important type of information collected by sensors. At the same time, the IoT in ITS enables various elements in the transportation system to communicate with each other, as well as with centralized systems. These elements include them but not limit to vehicles, pedestrians, and transportation infrastructures. With the increase in the number of connected infrastructures, transportation infrastructures now have the ability of coordinating with edge cloud and thus with a centralized operation instruction system. Then, the information is propagated back to vehicles via various kinds of sensors. The information then is transmitted among vehicles on the road with V2V connections. A graphical demonstration of this system is shown in Figure 1. With this information, vehicles are able to accomplish tasks that weren't possible before. For example, the automatic license plate detection and recognition is an application that has great potentials. With these tasks, ITS is becoming a crucial part in the ever-expanding IoT, which has the potential to improve the life of people significantly.

The vehicle ownership in many large and medium-sized cities in developing countries continues to grow. Urban traffic congestion problem has become increasingly prominent [1], which brings difficulties to urban management and affects the sustainable and healthy development of the city. Recognizing the vehicles in the transportation network is the first and most fundamental step to enable the automatic management of vehicles. Therefore, it is necessary to continuously use new technologies to improve urban traffic problems, make better use of existing road resources more efficiently, and continuously improve the smart city traffic system. V2V can help us reach this goal.

ITS [2] aims to effectively combine advanced technologies such as telecommunications, electronic information, and automation in recent years, combine with the corresponding traffic management architecture, to establish a system that can be accurate and effective in the traffic management.

Manuscript received 25 June 2021; revised 25 November 2021, 24 April 2022, 1 August 2022, and 1 September 2022; accepted 21 September 2022. Date of publication 20 October 2022; date of current version 5 December 2022. The Associate Editor for this article was S. H. A. Shah. *(Corresponding author: Sumin Li.)*

The authors are with the School of Information Engineering, Minzu University of China, Beijing 100081, China (e-mail: amycun@muc.edu.cn; smli@muc.edu.cn; ruixiang-0822@163.com; 2012048@muc.edu.cn).

Digital Object Identifier 10.1109/TITS.2022.3213018

1558-0016 © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DE GOIAS. Downloaded on September 25,2024 at 23:09:09 UTC from IEEE Xplore. Restrictions apply.

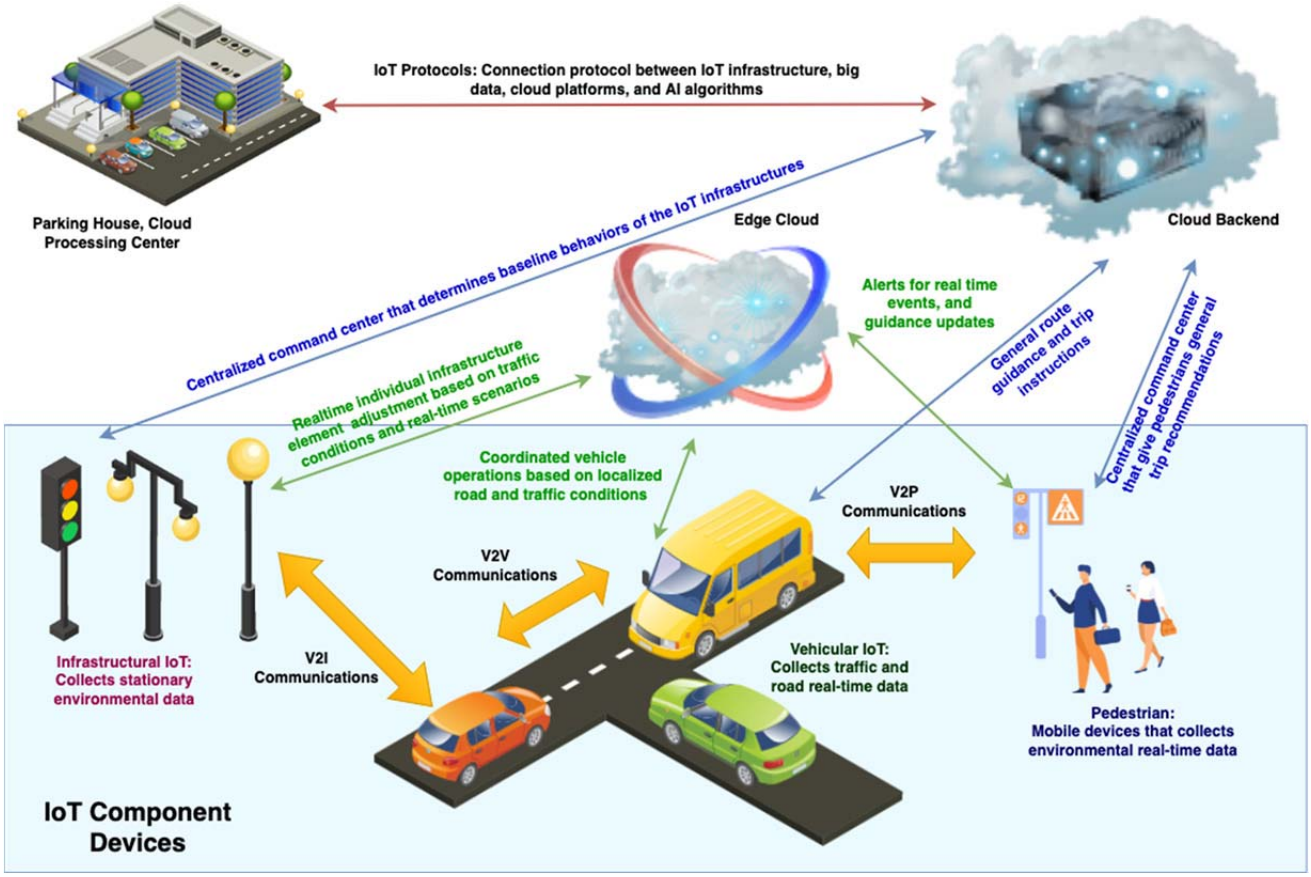


Fig. 1. A graphical demonstration of IoT-enabled ITS with V2V, V2I and V2P connections.

Compared with the traditional transportation system, the most significant feature of ITS is the combination of intelligence and transportation system. Integrating these technologies into the infrastructure of the traffic system, through the automatic recognition of illegal vehicles, can ultimately alleviate traffic congestion, ensure smooth traffic, reduce illegal driving, and improve safety. With vehicles equipped with sensors and communication modules, they can become active members of IoT, thus increase the intelligence of the overall ITS system.

Different from the traditional network system architecture, IoT is a network system that promotes the effective connection between things and things, and between things and the network, according to the standards and requirements of the protocol, and realizes the information exchange and communication [3]. In the beginning, IoT is mainly applied to simple device interconnection in a small area. With the increasing advancement of technologies, IoT is beginning to be used in many institutions and companies in a large area. IoT is equivalent to a mobile platform, a portable sensor and a camera in the highway, using high performance, low-power in-vehicle platform through image processing and machine learning, can monitor traffic information in each area. IoT manages network through the intelligent monitoring and processing [4]. 5G and AI can realize ultra-high-speed, low-latency collection and analysis on data collected by IoT. The data information collected by IoT can be transformed into a language that is easier for humans to understand, and the information can be transmitted to users more intuitively, and a more comfortable user experience can be established. Nowadays, most vehicles are equipped with cameras, which can perceive and monitor the surrounding environment in real-time. At the same time, the infinite infrastructure that the vehicle is equipped with can also make V2V communications. In V2V communications, there are many scenarios that require a high accurate detection and recognition for license plates.

License plate recognition system [5] has an important role in traffic management. Traditional license plate recognition relies on fixed cameras and discovers traffic accidents manually by operators. However, researches have shown that after more than 20 minutes of continuous view of video screens, and human eyes will not be able to pay attention to 90% of the screens, so manually monitoring has a big challenge. Highway patrol usually follows a certain path in preset patrolling areas. Once a traffic accident occurs, due to the congestion caused by the accident, the patrol vehicle cannot reach the site quickly. Moreover, the road management requires the use of magnetic induction coils and radars in some traffic information, which will be expensive, more troublesome, difficult to construct. Therefore, relying on human for license plate detection and recognition is slow, delayed and can have many false negative results.

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DE GOIAS. Downloaded on September 25,2024 at 23:09:09 UTC from IEEE Xplore. Restrictions apply.

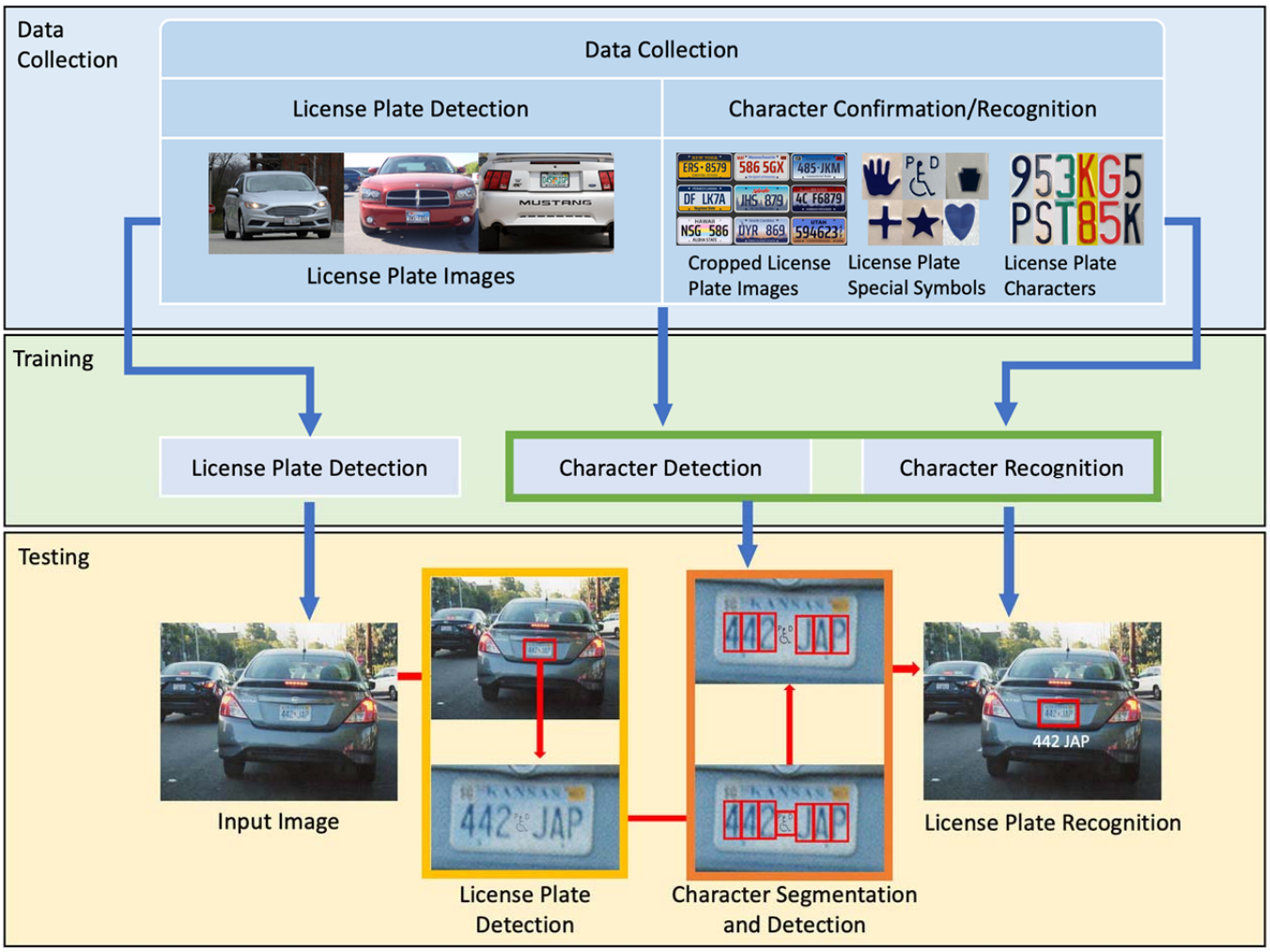


Fig. 2. General procedure of the license plate detection and recognition system.

However, with the help of AI, license plate can be detected and recognized with a better speed and accuracy. It generally follows a typical machine learning and computer vision framework. A graphical demonstration of such a system is shown in Figure 2. The first step is data collection. Here two kinds of data are collected: images with license plates for location, and images with license plate number and letters. With appropriate amount of collected data, we train license plate detection model, character detection and recognition model separately. Then with the models of the whole process properly trained, we test the workflow on images, first to detect the license plate, locate the letter/numbers and then recognize them. Johnson et al. [6] proposed an automatic license plate recognition system, constructing template, extracting features, and recognizing character. Lotufo et al. [7] used optical character recognition to manipulate the original image. First, they used the binary image to find the position of the license plate, and then the model recognized the boundary of the license plate to obtain features of character. Then they used the nearest neighbor classifier in the character library to compare with the character in it, extract the spare license plate numbers, check these numbers, and finally determine whether it is a real license plate. License plate detection and recognition has been widely applied to shopping malls, companies, and communities. However, only relying on fixed cameras is no longer enough to provide license plate information for all scenarios.

The IoT can achieve information interaction and visualization through the interconnection between the object, and the basis of the license plate detection and recognition must realize the digitization and visualization of the license plate. Therefore, some technologies of the IoT can be applied to the license plate detection and recognition system, which enables the originally independent parking space to access the Internet system to reach the target of the license plate detection and recognition. Vehicles will act as individual nodes in the IoT, and they can share license plate information after they are detected and recognized in the network via V2V connection.

The IoT has many types, including wireless technology such as ZigBee, Wi-Fi, Sigfox, Lora, and NB-IoT. The license plate detection and recognition system should be required to be available in a network, power consumption, and bandwidth do not require too high. Low-Power Wide-area Network (LPWA) is designed for this type of the IoT. Among them, NB-IoT

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DE GOIAS. Downloaded on September 25,2024 at 23:09:09 UTC from IEEE Xplore. Restrictions apply.

and Lora are the most developed LPWA technologies that develop prospects. Lora is an unlicensed spectrum, but the NB-IoT has been vigorously developed in some scenarios. The license plate detection and recognition system are also mainly using the NB-IoT. Intelligent recognition and data collection of object information is the premise of the application of the IoT in the license plate detection and recognition. The NB-IoT enables road cameras achieve automatic sensing and information collection, and directly uploads information to the information management platform. Compared to traditional wireless technologies, the NB-IoT has a wide range of coverage, large connection, and low cost, etc., which makes it advantage in the license plate detection and recognition. Integrating of the IoT into the license plate detection and recognition will be a new generation system.

At present, the issue of an automatic license plate detection and recognition has gradually evolved into a very important subject in the field of traffic engineering, and has received extensive attention from related scholars. This system covers a variety of modern technologies and is also the most important part of the ITS. This system mainly refers to the technology that can detect vehicles in the target area, accurately extract the characters and color of the license plate. In addition, it takes diversified technologies such as digital image processing and pattern recognition as the core, comprehensively analyzes and processes the images obtained by the device, and then extracts the license plate number to recognize the target vehicle.

Intelligent transportation based on computer vision is the fusion of a number of high and new technologies. Its key modules involve video image collection, target tracking, behavior recognition, high-performance computing, AI, etc. Through the integration of the IoT, a vehicle can search for other vehicles around it through wireless channels and form a self-organizing network with it when the vehicle is running. The controller inside the network will share various information. When abnormal driving conditions occur, radio signals can be used to warn and remind vehicles within the networked range, so that drivers can be alert in time and take actions to avoid serial accidents. At present, vehicles are mostly equipped with cameras that can perceive and monitor the environment in real-time. Also, the devices that the vehicle is equipped with can also make the vehicle communicate with each other. Under such a scenario, the automatic license plate detection and recognition will be the basis for realizing this function.

The license plate detection and recognition can automatically record and verify the license plate information of the vehicle. In recent years, the license plate detection and recognition has been widely used, and its application fields mainly include vehicle verification, traffic enforcement, high-speed toll, and parking management. At the same time, with other processing methods, the license plate detection and recognition can support a variety of functions, including vehicle positioning, electronic police and multi-channel payment. In addition, it is also an important basis for vehicle detection and vehicle scheduling. In recent years, image collection and image recognition technologies have been further developed, laying a good foundation for the wide application of the license plate detection and recognition.

In this paper, we propose an Internet-based license plate detection and recognition technology, and its implementation. The main task is to adapt to the rapid movement and complex road situation, by proposing the license plate detection and recognition based on a hybrid deep learning model by fusing YOLOV3 and CRNN. The main contributions of this paper are as follows: This paper studies the license plate detection and recognition issue, and proposes a hybrid deep learning model based on the fusion of YOLOV3 and CRNN. This model utilizes YOLOV3's fast detection rate, and CRNN's accurate recognition ability. We demonstrate this proposed model on an ensemble dataset created by merging a few existing datasets, and results prove that this model outperforms some advanced baseline models.

This paper is organized as follow: in section I, we discuss the background, motivation and contribution; then, in section II, the related work in this field is introduced from two different deep learning models including CRNN and YOLOV3; in section III, we propose a hybrid deep learning model by fusing YOLOV3 and CRNN for the license plate detection and recognition; in section IV, the dataset we used in this paper is introduced and constructed; and then this proposed hybrid deep learning model is trained and tested on the constructed dataset; in section V, we compare the proposed hybrid deep learning model with some baseline methods and also discuss the results; in section VI, we conclude the findings and discuss the future work.

II. Related Work

The goal of this paper is to detect and recognize the license plate on the road, so advantages and disadvantages of the target detection and recognition algorithm are particularly important. There are many scholars have attempted to solve similar issues in recent years and achieved some promising results, but they either are somewhat tangential to our issue or the results are not completely satisfactory [8], [9], [10]. The hybrid deep learning model proposed in this paper has two main parts including YOLOV3 and CRNN, so this section will introduce these two important parts.

A. YOLOV3 for Target Detection

You Only Look Once (YOLO) [11] is a CNN network based on one end-to-end network, to solve the target detection as a regression problem. This paper will use the YOLOV3 as a part of the target detection algorithm. YOLOV3 uses DarkNet-53 as the backbone structure of the network, a structure the authors specifically proposed for YOLOv3. It has fewer computations than RESNet-152, but performance is basically consistent. Its feature extraction capability and learning ability are stronger than others series in YOLO. YOLOV3 uses features pyramid network in multi-scale prediction tasks, which can predict three different scales of moving objects by selected anchor mechanisms, with a large number of resolutions with better positioning information. YOLOV3 uses binary cross entropy in terms of category prediction instead of softmax. Thus, solving the label category may have a duplicate issue, and the accuracy will not decline. The target function of the

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DE GOIAS. Downloaded on September 25,2024 at 23:09:09 UTC from IEEE Xplore. Restrictions apply.

anchor frame in YOLOV3 still uses a mean square error, while the other part of the target function is used to use binary cross entropy, which is used for two categories.

B. CRNN for Target Recognition

A common traffic camera requires the specific location of the moving vehicle to be detected in the highway, which involves the target detection technology. Because the road traffic is constantly changing, and due to the influence of weather, uneven road, outdoor light change, it is necessary to select a suitable robust target detection algorithm. Traditional target detection algorithms, such as background differential methods, interframe differential methods, etc., it is difficult to achieve motion vehicle detection in dynamic background. Machine learning algorithms, such as vehicle detection methods based on histogram of oriented gradients (HOG) and support vector machine (SVM), relatively low efficiency, and is prone to mishandling, as well as some of the deep learning methods that are proposed in recent years [12], [13]. Through the analysis of the current target detection algorithm, and comparison of various technologies on major target detection competitions, it can be found that the superior performance in the field of target detection is almost all in-depth learning techniques. Under the conditions of the environmental changes, considering target detection accuracy and real-time, the vehicle target detection algorithm based on deep learning is selected. For example, HOG, local binary mode (LBP) [14], and SIFT feature descriptors, etc., the target detection algorithm that requires manual feature extraction is always dominated in the past years. Moving on from the traditional machine learning methods, CRNN [15] is proposed for learning better image representation, especially for images with obvious spatial contextual dependencies and gained popularity in the various detection tasks [16], [17], [18]. In this paper, we utilize the CRNN proposed in [26] to recognize license plates. It took full advantages of CNNs that can effectively extract local features, and proposed CRNN. CRNN first utilizes CNN to extract local high-level correlation features of a path, and then feeds the correlation features into recurrent neural network to model the path representation. Its architecture is adaptable to obtain not only local features but also global sequential features of a path. To train and test the target detection algorithms, there are three famous public datasets created for the target detection: PASCAL VOC [19], ILSVRC [20] and MSCOCO [21]. Scholars mostly use these data sets to train their target detection models. In this paper, we will adopt the PASCAL VOC data set's format as input image, and our CRNN part of the model benefits from transfer learning of the original CRNN model, which is trained on the ILSVRC dataset.

III. Model Architecture

To solve the issue of the license plate detection and recognition, we propose a hybrid deep learning model by fusing YOLOV3 and CRNN to detect and recognize the license plate. The hybrid deep learning model inherits excellent object detection performance of YOLOV3, which can efficiently and accurately locate the license plate in the image. At the same time, the hybrid deep learning model also inherits an excellent performance of CRNN in the license plate recognition.

The hybrid deep learning model can directly use images as the input, avoiding the computational cost of extracting target candidate regions in advance. After processing, the final input contains the location, category and corresponding confidence of the target object. Among them, the returned value is in the form of a vector. First, the input image is separated into an $S \times S$ grid, and the subsequent search will be conducted in each grid. Then we set a number B, which represents is the number of bounding boxes that a grid can predict. The model will output a list of bounding boxes along with the predicted classes for each grid, which is represented by the corresponding confidence of each class in each bounding box. The confidence value reflects the probability of target detection, which is the probability of an object with a specific class appearing in a specific grid. The confidence is calculated as: $confidence = \Pr(Object) \times IoU$ where, $\Pr(Object)$ represents the probability of the bounding box output by the network contains an object from target class. For each bounding box in the grid, 5 values will be predicted accordingly: x, y, w, h, confidence. (x, y) represents the position of top left bounding box corner, w and h represent the width and height of the bounding box. The above probability meets the conditional probability model, which can be defined as $\Pr(Class_i|Object)$. This probability represents object from the i-th class falling into a specific bounding box. Define C to be the number of classes, while the number of bounding boxes B has no effect on C, the output of the network will be a one-dimensional vector representing the probability of B bounding boxes and C classes. The overall dimensional vector is:

$$dim = S \times S \times (B \times (5 + C)) \tag{1}$$

For each grid, the probability distribution of the object in the grid belonging to a particular class is calculated. In order to improve the accuracy, a threshold will be set to eliminate bounding boxes with lower confidence. Then we use the Non-Maximum Suppression (NMS) algorithm to remove unnecessary redundant bounding boxes, reducing the influence of unstable values on the stability of the final output result, and output the final target result.

When capturing a license plate on a highway, because it is impossible to ensure that all vehicles are in a relatively reasonable position in the camera, there may be excessive offset, or a certain deformation, and the vehicle may be in a high-speed driving status. Therefore, when the license plate occupies a small proportion in the target image, the following issues may also exist:

(1) The amount of information presented by the pixels in the corresponding detection region is limited, which leads to some more general target detection algorithms that are not effective in small object detection scenarios.

(2) During training, the labeling of small objects is prone to deviation. When target objects are small, the error of the labeling will have a greater impact. To improve the detection network module, we expand the original network by 3 to 4 scales to deepen the detection. This change makes

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DE GOIAS. Downloaded on September 25,2024 at 23:09:09 UTC from IEEE Xplore. Restrictions apply.

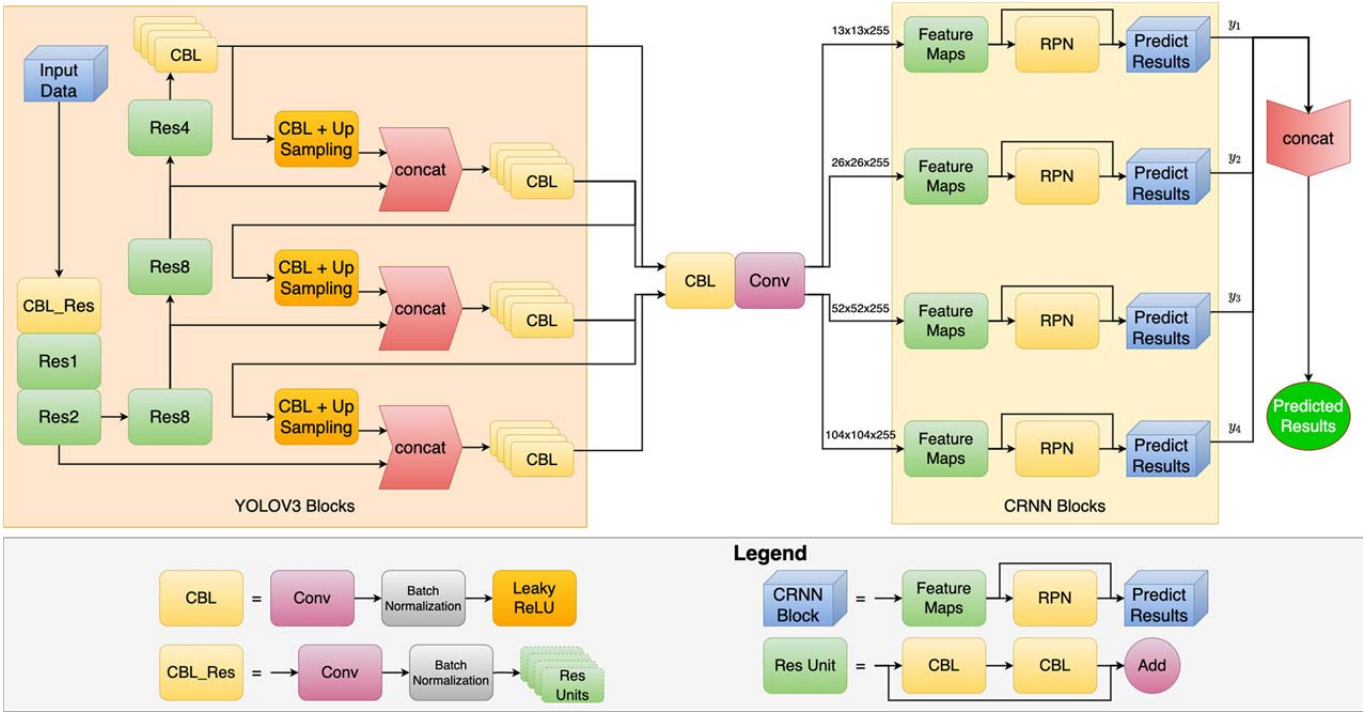


Fig. 3. The overall work flow of the proposed hybrid deep learning model.

the detection in smaller license plates more accurate, and the selection of anchor point frames is more accurate. It also enables the network itself to better integrate different fine-grained features in the higher and lower layers to perform multi-scale detection. Combined with the feature information with other features at different levels, the information amount contained in different feature levels has been improved.

There is no fully connected layer and pooling layer in the entire backbone network. In the forward propagation, the size transformation of the tensor is mainly realized by adjusting the step size of the convolution kernel. The design of its backbone network makes it more advantageous in feature extraction, thus promoting the performance of the network model. The network architecture of the hybrid deep learning model is shown in Figure 3. The network connects the general network of YOLOV3 with CRNN to replace the original, simpler standard prediction blocks.

As shown in Figure 3, the CBL is the combination of Convolution (Conv)+Batch Normalization (BN)+Leaky ReLU. BN and Leaky ReLU are inseparable and are the smallest components in the network. *Resn* (n represents a number) is the residual structure, which indicates how many residual units are in the residual block (res_unit), as shown in green in Figure 3, which is the largest component of the network. Concat stands for tensor concatenation, which connects the up sampling of the middle and later layers. This operation has a different meaning from the addition of the residual layer. Concatenation will increase the size of the tensor, while the addition of the residual layer is a direct addition and does not increase the number of tensors.

The entire network mainly uses up sampling to implement multi-scale feature maps. In Figure 3, the size of y1 to y4 is presented. Since there is a size difference between images, concat connects two tensors with the same size. To connect y2 and y3, we apply (2×2) up sampling to scale up y2 with siz $26 \times 26 \times 255$ eto the same size of y3. Similarly, to connect y3 and y4, we apply (2×2) up-sampling to scale up y3 with size $52 \times 52 \times 255$ to the same size of y4. This process expands the size of the tensor, increases the fine-grained detection, and improves the detection effect for small objects. Although this change increases the amount of network parameters and will reduce the detection speed, the improvement of feature extraction capabilities and model accuracy can make up for this loss.

In the next step, the license plate detected by YOLOV3 in the hybrid deep learning model is processed through CRNN. The detailed structure of CRNN is shown in Figure 4. It shares the convolutional layers at the end of YOLOV3 model, before generating feature maps. In this paper, we utilize a CRNN model that has a strong function and good performance [22]. Each column of the feature map corresponds to a rectangular region of the input image. The feature map is scanned column by column from left to right to generate the feature vector. CRNN arranges the rectangular region from left to right and contains the corresponding columns.

As shown in Figure 4, the network structure of CRNN consists of three parts: shared convolutional layer, region proposal network (RPN) tha's based on RNN, and the finalprediction part. The input image is processed by YOLOV3 and four branches are generated for object recognition, with the different scales of the feature map for each branch. Four branches

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DE GOIAS. Downloaded on September 25,2024 at 23:09:09 UTC from IEEE Xplore. Restrictions apply.

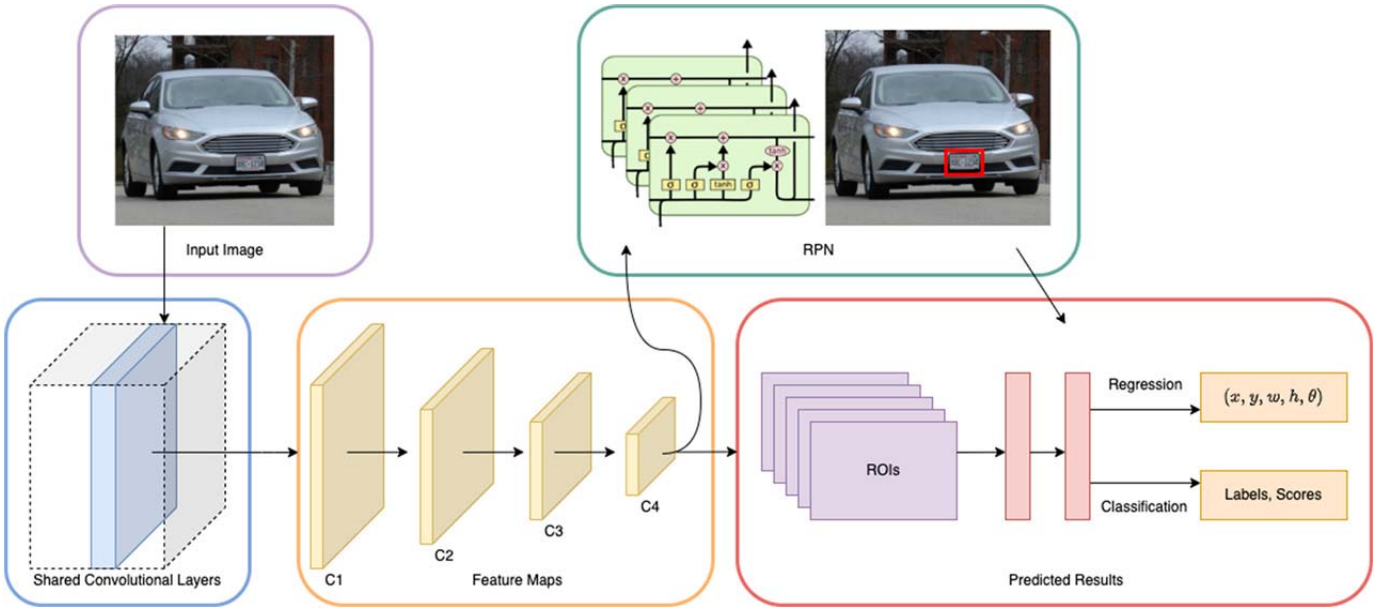


Fig. 4. The details of the CRNN part in the hybrid deep learning model proposed in this paper.

are input into the CRNN block to be further processed by the convolutional layers to generate feature maps for RPN. The RPN is an RNN that made of LSTM cells, and is established to predict each column of feature sequences. Finally, the RPN's output is used in the classification part to convert into ROIs and then classification results. The recurrent layer feeds back the error differential to the convolutional layer of the previous stage through back propagation, so that the recurrent layer and the convolution layer are jointly trained in a unified network.

A deep bidirectional recurrent neural network is built on top of the convolutional layer, called the recurrent layer. The loop layer has an ability to capture the context information. Compared with the independent processing of the characters in the license plate image, the use of context information to recognize the sequence increases a certain degree of stability. In CRNN, the depth feature is transformed into a sequence representation to maintain the invariance of the sequence object length. The feature sequence is $x = \{x_1, x_2, \ldots, x_T\}$, and the label distribution is $y_t, t \in \{1, 2, \ldots, t\}$ of each dimension.

Traditional RNN that are usually used to extract context to make predictions has vanishing gradient issue. To battle this issue, long short-term memory (LSTM) units are used in our RPN part. Assume the sequence feature x_t is input to each LSTM unit in an orderly manner, and the output C_t will be used as the input of the next LSTM unit. Preprocess marker sequences with different lengths, the input sequence x_t in each unit is preserved and updated through the sigmoid function. The design of LSTM allows it to capture the long-term relationship in the image sequence, and convert the hidden state of the LSTM into the probability of each character classification through the Softmax layer classification.

The last stage of the license plate recognition based on CRNN is to predict characters for transcription. For a character sequence $x = \{x_1, x_2, \ldots, x_t\}, x_t \in R^{Char \cup Emp}$, set $Char$ represents all characters, set Emp represents empty characters, and the sequence features encoded by x_t are transcribed by connectionist temporal classification (CTC) to form a sequence pat π. CTC is an algorithm commonly used in speech recognition, text recognition and other applications to solve the issue that the input and output sequences have different lengths and cannot be aligned. Then the probability path of π is:

$$p(\pi \mid x) = \prod_{t=1}^{T} p(\pi_t, t) \tag{2}$$

In order to perform an accurate output, the trained model must maximize the probability of correct output in the distribution, and multiplication means that the probabilities of all characters of a path are multiplied. The total probability of the decoding path π can be calculated by adding the probabilities of all the corresponding sequences, which is defined as:

$$p(l \mid x) = \sum_{\pi : \mathrm{B}(\pi) = l} p(\pi \mid x) \tag{3}$$

The alignment rule of the transcription layer is that multiple inputs correspond to one output, and its conditional probability is the sum of the probabilities of the same path. The transcription layer provides a differentiable loss function through CTC for end-to-end training of the recurrent layer and the convolutional layer, which solves the issue of input and output sequence alignment.

IV. Model Training and Results

A. Datasets

After completing the overall license plate detection and recognition model, a large number of license plate datasets need to be used to train the model, so that the parameters can be optimized. When performing target detection, the required dataset is relatively special and has required specifications and formats. At present, most commonly used

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DE GOIAS. Downloaded on September 25,2024 at 23:09:09 UTC from IEEE Xplore. Restrictions apply.

TABLE I
SPECIFICS OF THE EXPERIMENTAL ENVIRONMENT

Item	Value
Operating System	Ubuntu 16.04
CPU	Intel® Xeon E5-2643 @ 3.4GHz
RAM	16 GB
GPU	NVDIA GeForce GTX 1080 Ti
Libraries	OpenCV 3.4.2
CUDA	8.0

image dataset formats includePASCAL VOC, ILSVRC and MSCOCO. In this paper, we use the PASCAL VOC format to create the datasets. In this paper, we are creating the dataset to verify this proposed model from public datasets for the following reasons. First, high-quality labeled datasets for the license plate detection and recognition are very scarce. Large models like YOLOV3 and CRNN we are using in this paper need large amount of data to train. Thus, we need to construct a dataset that is enough to train the model properly. Secondly, each of the sub-datasets is relatively similar, which means each dataset only represents one or a few scenarios. However, in real application of this proposed model, there are different lighting, weather, etc. conditions. Thus, using data from multiple datasets can benefit the model. In terms of license plate data collection, the training data mainly comes from several open-source license plate detection datasets [23], [24], [25], [26].

B. Training

In order to test the effectiveness of the proposed license plate detection and recognition model, 6000 images collected as the training set are used to implement related experiments. The specifics of the experimental environment for the proposed model are shown in Table I:

In the training process, in order to obtain better model effects, it is often necessary to adopt some strategic adjustments to parameters. The following four parameters are mainly adjusted: momentum factor, learning rate, learning rate decay factor, weight deca coefficient. Among them, the setting of the learning rate has a greater impact on the network training.

Learning rate: It mainly controls the update speed of the weights in the network model. Overly low learning rate will slow the network convergence, and overly high learning rate will make the model miss the optimal point, thus affecting the final optimization of the objective function. Learning rate decay factor: It can prevent the objective function from being unable to control the convergence speed and getting a wrong optimal value. Momentu factor: The strategy of "inertia" is added to the model optimization process to speed up the network learning and convergence. Weight deca coefficient: It has the function of preventing the model from over-fitting to the training data. The above-mentioned related parameters have a relatively large impact on the performance of the model. In the optimization process, the error curve can be used for control. If the model is not well-fitted, we should consider reducing the value of the weight decay coefficient appropriately. Based on the related parameter optimization, many experiments have been carried out. The finally selected parameters are shown in Table II:

TABLE II
SELECTION OF HYPERPARAMETERS FOR LICENSE PLATE DETECTION NETWORK

Learning rate	Learning rate decay factor	Weight decay coefficient	Optimizer	Momentum factor
0.01	0.1	0.0005	Adam	0.9

Except for the optimized parameters above, the rest of the parameters are fixed. The decay step is 40000, the batch size is 64, and the maximum iteration is 50000. The license plate detection is mainly to obtain the region where the license plate is located, and provide input parameters for the recognition model. The accuracy of the detection is closely related to the effect of the license plate recognition. We use the average Intersection over Union (IoU), which reflects the accuracy of the detection coordinates, to measure the effectiveness of the license plate detection. The larger the value of this index, the more accurate the detection and the better the effect. The scatter diagram of changes in related training parameters is shown in Figure 5. As shown in Figure 5, the training loss, average IoU, average confidence and average recall changes with the number of iterations during the training process are respectively shown. After 8,000 iterations, the parameters tend to be stable. The training loss drops to about 0.25, the average IoU stabilizes at 0.78, the average confidence converges to 1.0, and the average recall approaches 0.84. Considering the convergence of various parameters, the overall model training results are good.

C. Evaluation Metrics

In the field of the target detection, Accuracy, Precision, Recall, F-score, IoU, mean Average Precision (mAP), frame-per-second (FPS), etc. are usually used as metrics to evaluate the quality of the model. Recall and accuracy are generally proportional to the ratio of positive and negative samples, while recall and precision are often contradictory values. Systems with high recall tend to have low precision. On the contrary, a system with a higher precision may have a lower recall. To solve this issue, F-score is introduced, and the value is proportional to the stability and performance of the system. The IoU, also known as the overlap rate, is mainly used to measure the degree of overlap between the test results and the label box of the initial data. Its value is proportional to the quality of the test. mAP, where AP measures the quality of the trained model in each category. mAP is the mean value of AP of all categories, which measures the average quality of the model in all categories.

V. RESULTS AND DISCUSSION

In this section, we will use multiple metrics to compare and analyze traditional methods and hybrid deep learning-based method fo the license plate detection and recognition. Different methods are compared mainly from precision, recall, F-score,

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DE GOIAS. Downloaded on September 25,2024 at 23:09:09 UTC from IEEE Xplore. Restrictions apply.

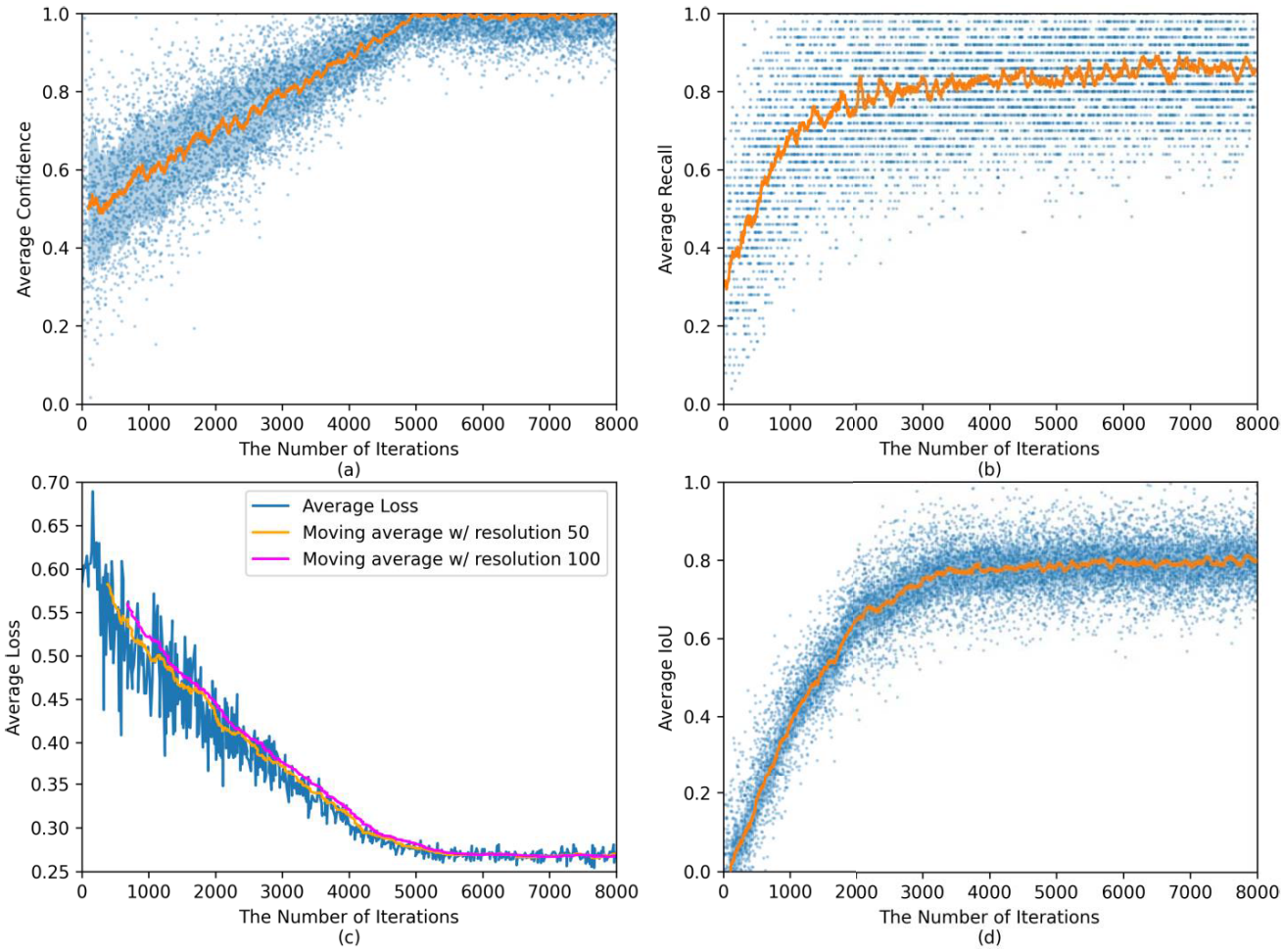


Fig. 5. Various experiment evaluation criteria and their results of the proposed model with the number of iterations: (a) average confidence (b) average recall (c) average loss (d) average IoU.

IoU, mAP, and FPS. In addition, we will also analyze the cases where our model does and does not perform well, and why they are performing this way.

First, we used OPENCV as a baseline method for comparison. For reference, OPENCV uses contours of the digits for this task. To do so, this process consists of several image processing steps including: resize images, conversion to greyscale, noise reduction, and binarization. The first three steps are fairly common. After the image preprocessing steps, the image becomes a black and white image with only contour lines. At the state, we then used OPENC's contour finding tools to find the contour. Each contour we found is approximated as a polygon, and we selected the ones with 4 edges. In this way, we can extract license plates from the images. Then we used character segmentation (CS) to extract the individual characters of the license plate. CS is a commonly used step in optical character recognition (OCR) tasks, so we will not go into details on it. Once we have the characters individually extracted, we then used standard computer vision technics to recognize the digits.

The overall test results of various license plate detection and recognition methods are shown in Figure 6. As shown in Figure 6, compared with the traditional method based on OPENCV computer vision, the detection and recognition

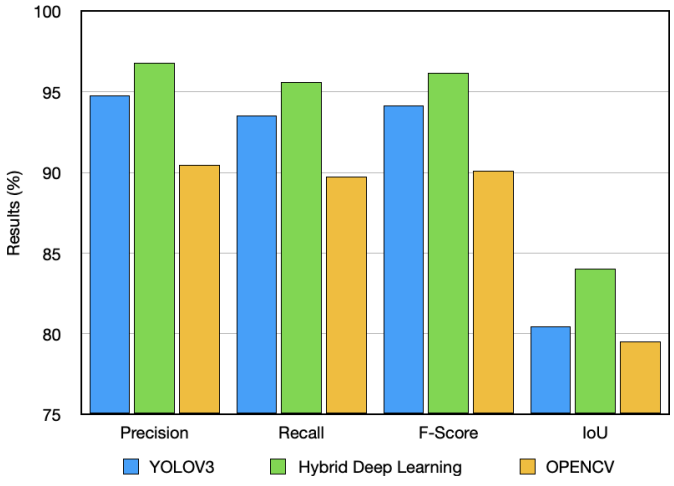


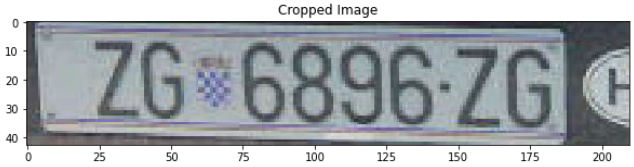
Fig. 6. Comparison of proposed method and other methods.

based on the hybrid deep learning model has better results. More specifically, the OPENCV method performs much worse compared to the YOLOV3 and our hybrid deep learning method. When looking at the cases that OPENCV doesn't perform well, we notice it has several disadvantages. First, its

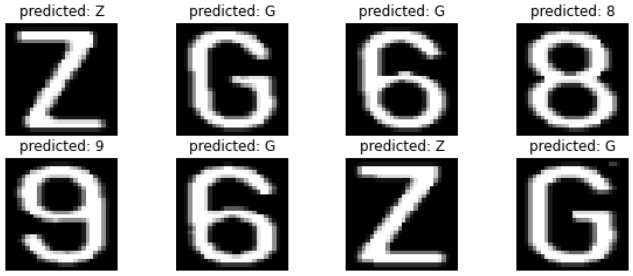
Authorized licensed use limited to: UNIVERSIDADE FEDERAL DE GOIAS. Downloaded on September 25,2024 at 23:09:09 UTC from IEEE Xplore. Restrictions apply.

ability to handle distortion is very poor. The OPENCV is trained using a standardized dataset that has rather good format. What this means is although there are numbers and digits in various formats, they don't have perspective distortion. However, in reality, the images in the dataset have perspective distortion: the photos are not always taken from the front at a right angle; but rather, they can be taken from the side, which will result in the license plate to be in a trapezoid, and the characters on the license plate will be distorted, no longer resembling the characters seen in the training dataset. In addition, the OPENCV method is also susceptible to similar characters. For instance, we notice many cases where the failed recognition is when the letter "G" was recognized as the number "6" and also the other way aroun; same happened occasionally for "7" and "1" depending on the font of the characters. These drawbacks of OPENCV are expected, since it's a rather simplistic method. The performance improves significantly for YOLOV3 model. We adapted the original YOLOV3 from [11], and then fine-tuned it on a license detection dataset. In this way, we can obtain a weight set of YOLOV3. Then using the trained YOLOV3, we can detect potential bounding boxes in the image. We will drop the bounding boxes that have low confidence. As a result, we can identify the bounding box of the license plate. Next, we are using CS again to separate the characters. In the CS step, we undergo the following steps: First, we resize the image to a size that all characters are distinct and clear. Then we convert the colored image to greyscale, and use a threshold to convert the greyscale image to binary image with only black and whit; and in this process, the threshold is 215. For pixels with density below the threshold, it is given value of 0, which correspond to white. Then we use eroding on the binary image, which is to remove unwanted pixels from the object boundary. With the image free of its noises, we define the size for our desired characters to selected the characters in the license plate, and are ready for extracting the characters. While with YOLOV3, it can achieve relatively good results, but it still underperforms when comparing to our proposed hybrid deep learning model. We speculate the reason is: YOLOV3 does well when detecting objects in the image. This means YOLOV3 network can detect the license plate bounding box well. However, its functionality stops after the detection. The results are then passed onto CS steps for the final character recognition. Its capability is limited by the capability of the character recognition model. Therefore, in the tuning of the models, we notice that compared to the OPENCV method, YOLOV3 is able to extract the bounding boxes of the license plate very well. However, the lacking of its performance happens when the CS steps misclassify the characters. A sample failure case for YOLOV3 is shown in Figure 7. Figure 7 (a) shows the processed image and the bounding box with the highest confidence detected by YOLOV3. Then, Figure 7 (b) shows the cropped license plate area of the bounding box. Lastly, Figure 7 (c) shows how each letter is isolated and predicted. As we can see, the prediction made mistakes for cases of letter "6", which is misclassified as "G".

(a) Processed photo with bounding box by YOLOV3 from [26]



(b) Cropped license plate area



(c) Characters after CS and their prediction values

Fig. 7. Misclassified license plate by YOLOV3 demonstration.

Therefore, to improve the YOLOV3 and OPENCV models, we propose our hybrid deep learning model in this paper. Our CRNN can significantly improve the performance of license plate recognition after YOLOV3 detects the location and bounding box of the license plate. Our model improves the overall results when compared to the YOLOV3 model. This benefit is achieved by incorporating multi-scale CRNN, so that the different scales of the feature map are processed individually. The hybrid deep learning model has higher precision, recall, F-score, and IoU, indicating that its robustness is better. Compared with the original YOLOV3, the performance of the hybrid deep learning model is also better. We also compare the hybrid deep learning model with both baseline methods in another two metrics including mAP and FPS.

As shown in Table III, the detection and recognition based on hybrid deep learning model is much faster than the OPENCV method, and the mAP is also improved. This is because this hybrid deep learning model including YOLOV3 has an optimized structure for faster training and inference on GPUs. Our proposed hybrid deep learning model has slightly better performance compared to YOLOV3 in terms of mAP, although a slightly higher computation time. The reason is by

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DE GOIAS. Downloaded on September 25,2024 at 23:09:09 UTC from IEEE Xplore. Restrictions apply.

TABLE III
PERFORMANCE OF HYBRID DEEP LEARNING VS BASELINE MODELS

Method	mAP (%)	Speed (FPS)	GPU
YOLOV3	94.758	46	GTX 1080 Ti
OPENCV	90.475	21.8	GTX 1080 Ti
Hybrid Deep Learning	96.784	38	GTX 1080 Ti

deepening the detection scale, the hybrid deep learning model complexity has increased, thus the running speed is slightly slower. In addition, compared to the original YOLOV3, this hybrid deep learning model also includes a CRNN component which greatly improves its ability in the license plate recognition. Although this hybrid deep learning model is 8 FPS lower than that of YOLOV3, from the perspective of the real-time detection and recognition, it is about 0.005 seconds slower per image compared to YOLOV3, which is an insignificant difference in reality. Considering the increase in mAP results, this slight speed decrease has a rather low impact.

Almost all IoT devices suffer from time issues due to limited computational resources, under the same condition, having a higher computation speed is important for the overall performance. However, by using different levels of fine-grained features for the detection, this improvement of mAP taken by this hybrid deep learning model is definitely important.

VI. CONCLUSION

Aiming at the existing issue of the license plate detection and recognition, this paper proposes a hybrid deep learning model by fusing YOLOv3 and CRNN to solve this issue. everaging the excellent object detection ability of YOLOV3, we expanded the original three-scale detection to four-scale detection. Then the detection features are input into a CRNN network for further refined license plate recognition. The CRNN network used in this paper consists of an RPN consists of LSTM cells, instead of the traditional RNN cells to mitigate the vanishing gradient problem that is common for RNNs. Then we trained and tested on an ensemble data set consist of multiple datasets. Through experimental comparison, the license plate detection model proposed in the paper has high accuracy, better comprehensive performance, and excellent robustness. The proposed model was also compared against several other models and demonstrated the superiority of our model. However, due to current limitation in dataset, we are only able to train and test our model on a relatively small dataset, and also in fixed application environment, that is limited by the environmental condition of the data set. In the future, we would like to test our algorithm on an actual vehicle in motion and observe its behavior. In addition, we are currently limiting to alphanumeric characters, using our model on other languages such as Chinese, Arabic, Korean will not work. Future work should look into expanding the applicability of this model.

REFERENCES

[1] T. Wu et al., "Multi-agent deep reinforcement learning for urban traffic light control in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 8243–8256, Aug. 2020, doi: 10.1109/TVT.2020.2997896.

[2] M. Veres and M. Moussa, "Deep learning for intelligent transportation systems: A survey of emerging trends," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 8, pp. 3152–3168, Aug. 2020, doi: 10.1109/TITS.2019.2929020.

[3] W. Iqbal, H. Abbas, M. Daneshmand, B. Rauf, and Y. A. Bangash, "An in-depth analysis of IoT security requirements, challenges, and their countermeasures via software-defined security," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 10250–10276, May 2020, doi: 10.1109/JIOT.2020.2997651.

[4] F. Zhu, Y. Lv, Y. Chen, X. Wang, and F. Wang, "Parallel transportation systems: Toward IoT-enabled smart urban traffic control and management," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 10, pp. 4063–4071, Oct. 2020, doi: 10.1109/TITS.2019.2934991.

[5] C. N. E. Anagnostopoulos, I. E. Anagnostopoulos, I. D. Psoroulas, V. Loumos, and E. Kayafas, "License plate recognition from still images and video sequences: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 3, pp. 377–391, Sep. 2008, doi: 10.1109/TITS.2008.922938.

[6] A. S. Johnson and B. M. Bird, "Number-plate matching for automatic vehicle identification," in *IEE Colloquium on Electronic Images and Image Processing in Security and Forensic Science*. Edison, NJ, USA: IET, May 1990, p. 4.

[7] R. A. Lotufo, A. D. Morgan, and A. S. Johnson, "Automatic number-plate recognition," in *IEEE Colloquium on Image Analysis for Transport Applications*. Edison, NJ, USA: IET, 1990, p. 6.

[8] A. H. Ashtari, M. J. Nordin, and M. Fathy, "An Iranian license plate recognition system based on color features," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 4, pp. 1690–1705, Aug. 2014, doi: 10.1109/TITS.2014.2304515.

[9] X. Fan and W. Zhao, "Improving robustness of license plates automatic recognition in natural scenes," *IEEE Trans. Intell. Transp. Syst.*, early access, Feb. 24, 2022, doi: 10.1109/TITS.2022.3151475.

[10] H. Xu, X.-D. Zhou, Z. Li, L. Liu, C. Li, and Y. Shi, "EILPR: Toward end-to-end irregular license plate recognition based on automatic perspective alignment," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 3, pp. 2586–2595, Mar. 2022, doi: 10.1109/TITS.2021.3130898.

[11] J. Redmon, S. Divvala, and R. Girshick, "You only look once: Unified, real-time object detection," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 779–788.

[12] M. Woźniak, J. Siłka, and M. Wieczorek, "Deep neural network correlation learning mechanism for CT brain tumor detection," *Neural Comput. Appl.*, to be published, doi: 10.1007/s00521-021-05841-x.

[13] J. Sang et al., "An improved YOLOv2 for vehicle detection," *Sensors*, vol. 18, no. 12, p. 4272, Dec. 2018.

[14] G. Zhang, X. Huang, and S. Z. Li, "Boosting local binary pattern (LBP)-based face recognition," in *Proc. Chin. Conf. Biometric Recognit.* Berlin, Germany: Springer, 2004, pp. 179–186.

[15] Z. Zuo et al., "Convolutional recurrent neural networks: Learning spatial dependencies for image representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2015, pp. 18–26, doi: 10.1109/CVPRW.2015.7301268.

[16] N. Kousik, Y. Natarajan, R. Arshath Raja, S. Kallam, R. Patan, and A. H. Gandomi, "Improved salient object detection using hybrid convolution recurrent neural network," *Expert Syst. Appl.*, vol. 166, Mar. 2021, Art. no. 114064.

[17] F. Baghaei Naeini, D. Makris, D. Gan, and Y. Zweiri, "Dynamic-vision-based force measurements using convolutional recurrent neural networks," *Sensors*, vol. 20, no. 16, p. 4469, Aug. 2020.

[18] J. A. Chamorro Martinez, L. E. Cué La Rosa, R. Q. Feitosa, I. D. Sanches, and P. N. Happ, "Fully convolutional recurrent networks for multidate crop recognition from multitemporal image sequences," *ISPRS J. Photogramm. Remote Sens.*, vol. 171, pp. 188–201, Jan. 2021.

[19] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and W. Zisserman, "The PASCAL visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Sep. 2010, doi: 10.1007/s11263-009-0275-4.

[20] O. Russakovsky et al., "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015, doi: 10.1007/s11263-015-0816-y.

[21] T. Y. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 740–755.

[22] Q. Zhu, X. Zhou, J. Tan, and L. Guo, "Knowledge base reasoning with convolutional-based recurrent neural networks," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 5, pp. 2015–2028, May 2021, doi: 10.1109/TKDE.2019.2951103.

[23] V. Ribeiro et al., "Brazilian mercosur license plate detection: A deep learning approach relying on synthetic imagery," in *Proc. IX Brazilian Symp. Comput. Syst. Eng. (SBESC)*, Nov. 2019, pp. 1–8, doi: 10.1109/SBESC49506.2019.9046091.

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DE GOIAS. Downloaded on September 25,2024 at 23:09:09 UTC from IEEE Xplore. Restrictions apply.

[24] *Real-Time Auto License Plate Recognition With Jetson Nano*. Accessed: Dec. 2020. [Online]. Available: https://github.com/winter2897/Real-time-Auto-License-Plate-Recognition-with-Jetson-Nano/blob/main/doc/dataset.md
[25] *Number Plate Datasets*. Accessed: Jan. 2021. [Online]. Available: https://platerecognizer.com/number-plate-datasets/
[26] R. Laroca, E. Cardoso, D. Lucio, V. Estevam, and D. Menotti, "On the cross-dataset generalization in license plate recognition," in *Proc. 17th Int. Joint Conf. Comput. Vis., Imag. Comput. Graph. Theory Appl.*, 2022, pp. 1–13.

Xiuqin Pan received the B.E. degree in electrical technology and the M.E. degree in power system and automation specialty from Zhengzhou University, Zhengzhou, China, in 1994 and 1999, respectively, and the Ph.D. degree in control theory and control engineering from the Beijing Institute of Technology in 2002. She is currently a Professor with the School of Information Engineering, Minzu University of China. Her current research interests include parallel algorithm and intelligent systems.

Sumin Li received the M.S. degree in testing technology and automatization equipment from the School of Electrical Engineering, Zhengzhou University, China, in 2003. She is currently an Instructor with the School of Information Engineering, Minzu University of China, Beijing, China. Her main current research interests include big data, intelligent computing, and intelligent information processing and systems.

Ruixiang Li received the M.S. degree in basic mathematics from the School of Minzu University of China, Beijing, China, in 2013. He is currently an Experimenter with the School of Information Engineering, Minzu University of China. His main current research interests include intelligent computing and intelligent information processing and systems.

Na Sun received the Ph.D. degree in signal and information processing from the Beijing University of Posts and Telecommunications in 2010. She is currently an Associate Professor at the School of Information Engineering, Minzu University of China. Her current research interests include parallel algorithm and intelligent systems.

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DE GOIAS. Downloaded on September 25,2024 at 23:09:09 UTC from IEEE Xplore. Restrictions apply.