

ResNet and LSTM Based Accurate Approach for License Plate Detection and Recognition

Naaman Omar

Information Technology Department, Amedi Technical Institute, Duhok Polytechnic University, Duhok 42001, Iraq

Corresponding Author Email: naaman.omar@dpu.edu.krd



<https://doi.org/10.18280/ts.390514>

ABSTRACT

Received: 25 June 2022

Accepted: 15 October 2022

Keywords:

license plate detection and recognition, ResNet, deep features, LSTM classification

The identification and recognition of automatic license plates (ALP) are critical for traffic surveillance, parking management, and preserving the rhythm of modern urban life. In this paper, a deep learning-based method is proposed for ALP. In the proposed work, the license plate region is initially segmented in a given vehicle image, and the plate number and city region are extracted from the segmented license plate region. Residual neural networks (ResNet) architecture-based deep feature extraction is considered. The fully connected layer of the ResNet model is used to obtain the deep features for the cropped Arabic numbers and city regions, respectively. The extracted features are fed into the sequential input layer of the Long Short-Term Memory (LSTM) classifier. Various experiments are carried out on a dataset that was collected in the northern Iraq region and the classification accuracy score is used for performance evaluation. According to the obtained results, the proposed method is effective where the calculated accuracy scores were 98.51% and 100% for Arabic numbers and city regions, respectively. The performance comparison of the proposed method with some of the existing methods indicates the high performance of the proposed study.

1. INTRODUCTION

For today's modern transportation networks, ALP detection and identification are crucial. ALP recognition has the potential to be utilized in a range of applications, including traffic surveillance, security-related uses, and vehicle parking management since it can be used to identify cars from their license plates [1, 2].

Various studies on ALP detection and recognition have been conducted over the last two decades. For example, Kakani et al. [3] present an algorithm that employs artificial neural networks (ANN) to enhance ALP detection and recognition. The authors evaluated the performance of their proposed method using 300 vehicle images, achieving accuracies of 96.7 percent for detection and 92.2 percent for recognition. An algorithm for ANP detection and recognition was proposed by Prabhakar et al. [4]. The authors first transformed the input image from color to grayscale, then used median filtering to improve the image quality. The Hough transform was used to detect the edges and local patterns in the plate region after preprocessing. The binary image was then enhanced with morphological operations to detect the exact location of the license plate. One hundred fifty color images taken in various conditions were used to assess the performance, with a detection accuracy score of 94.0 percent and a recognition accuracy score of 92.0 percent. Singh et al. proposed an algorithm for locating number plates and recognizing characters [5]. Gaussian filtering and morphological operations were used to improve the input images, with character-positioning being used to locate the license plate. For character recognition, a support vector machine (SVM) classifier was used. A total of 419 images were used to evaluate the performance, all of which were taken under various conditions. The proposed algorithm had a

success rate of 97.21 percent for plate location and 95.06 percent for character recognition. Li et al. proposed an end-to-end CNN model for detecting and recognizing license plates [6]. In a single forward pass, they built a unified DNN model that localized license plates and recognized letters simultaneously. The authors evaluated the performance of their proposed method using three datasets. The detection and recognition scores obtained were both above 99.0%.

For Brazilian license plates, Montazzolli and Jung [7] proposed a deep learning-based end-to-end ALP detection and recognition system. Deep CNN architecture underpins the developed system. The authors chose to use a Brazilian license plate dataset that was made available to the public. With detection and recognition scores of 99.0 percent and 93.0 percent, respectively, their proposed system was able to correctly detect and recognize all seven characters on the Brazilian license plate. For license plate recognition, Cheang et al. [8] proposed a hybrid model that combined a CNN and a Recurrent Neural Network (RNN). The authors used CNN for feature extraction, and RNN was used to classify the extracted features. They used a dataset of 2,713 labeled license plate images in their experiments, and they got a score of 76.53 percent accuracy. Zang et al. [9] proposed a method for ALP detection and recognition that used deep learning and visual attention that was focused on detecting and recognizing Chinese car license plates. For the detection of license plates, a modified visual attention model was chosen. CNN and SVM classifiers were used to recognize Chinese characters after character segmentation. Even under the conditions of changing illumination and noise contamination, the authors claimed that the proposed scheme produced better results.

Bulan et al. [10] proposed a non-segmented and non-annotated ALP detection and recognition method. Using a two-stage approach, the authors first located the license plate

region. A CNN classifier was used to determine the location of the license plate in a group of candidate license plate regions. Following that, character recognition was performed using a probabilistic inference method based on the hidden Markov model (HMM). Experiments were carried out on a variety of images captured in realistic conditions, with promising results. For ALP recognition, Shivakumara et al. [11] developed a method based on CNN and Bi-Directional Long Short Term Memory (BLSTM). The authors chose to use a CNN model for deep feature extraction, with the BLSTM method for classification. In their experiments, they used a publicly available license plate dataset and achieved a recognition accuracy of 90.51 percent. Yang et al. [12] proposed a novel method for Chinese LPR that used a kernel-based Extreme Learning Machine (ELM) classifier and deep feature extraction. A pretrained CNN model was used to extract deep features, and the experimental results were superior to other CNNs that used Softmax or SVM. For ALP detection in complex backgrounds, Kim et al. [13] proposed a two-step deep approach. The authors detected the car region within a given image in the first step and then determined the license plate region within the previously detected car region in the second step. To generate multiple candidate license plate regions, the faster regional CNN was combined with a hierarchical sampling method in the first step. A CNN architecture was used to remove the non-plate regions. Experiments on the Caltech dataset revealed a 98.39 percent precision rate. Selmi et al. [14] developed a deep learning-based ALP detection and recognition system. Plate detection, character segmentation, and character recognition were the three steps of the proposed system. For image quality improvement, the authors used various preprocessing steps such as noise filtering and contrast enhancement. For plate detection, CNN was used, and image segmentation was used for character extraction. For character recognition, an additional CNN model was considered, and the results were improved.

Yaseen et al. [15] proposed a method for license plate detection. The authors used a cascade HOG feature descriptor with (AdaBoost) technique to detect a region of interest from the vehicle image. The authors used a vehicle dataset from north Iraq for training and testing processes. The accuracy rate for this model was 89.66%. Wang et al. [16] presented a system for LPR that used a multi-task convolutional neural network. The authors detected the license plate by using MTCNN and then recognize it using end-to-end methods. The precision achieved by this method was 98% after being applied to the Chinese City Parking Dataset. Henry et al. [17] proposed a method to recognize vehicle LP using YOLO deep learning. The dataset used in this work contains a multinational license plate. The accuracy achieved acceptable results for 17 countries LP. Izidio et al. [18] proposed a method for Brazilian LP detection and recognition. Tiny YOLOv3 and CNN were used for license plate detection from vehicle images and then character recognition. The method used real vehicle images under different weather conditions. The achieved success rate was 99.37% and 98.43% for detections and recognition, respectively. Dong et al. [19] proposed a method for LPR through two stages. The authors used R-CNN and a cascade of fast region proposal networks for LP detection in the first stage. In the second stage, the authors for LP recognition use a parallel spatial transform network. This method was evaluated on the Chinese LP dataset and achieved good results.

As the reviewed literature was considered, it was observed

that each work has its advantages and disadvantages. Traditional machine learning and deep learning approaches have been proposed for various LP detection problems. Besides, there have been a limited number of studies on LP detection where Arabic numbers and alphabets are on them. Thus in this study, a deep learning-based method for ALP is suggested in this study. The plate number and city region are retrieved from the segmented license plate region in a particular vehicle picture after it has been segmented. Deep feature extraction using residual neural networks (ResNet) architecture is examined. The deep features for the chopped Arabic numbers and city areas are obtained using the ResNet model's fc1000 layer. The retrieved features are 1000 dimensional, and they are sent into the Long Short-Term Memory (LSTM) classifier's sequential input layer. Several tests are conducted on a dataset gathered in northern Iraq, and the classification accuracy score is used to evaluate performance. According to the collected data, the suggested approach is practical with estimated accuracy scores of 98.51 percent and 100 percent for Arabic numerals and city regions, respectively.

The paper is organized as follows. In the next section, the proposed method and the related theory are given systematically. In Section 3, the experimental works and the results are discussed. The paper is concluded in Section 4.

2. PROPOSED METHOD

Figure 1 shows a schematic illustration of the proposed study. As seen in Figure 1, the initial step of the proposed work is the detection of the license plate region. A deep semantic segmentation approach namely SegNet is employed [2, 20]. The considered license plates have three distinct regions, as shown in three different colors. These regions are plate number region, city region, and country region, respectively. After segmentation of the license plate region, the plate number region and the city region are saved for further processes, and the country region is discarded as it is identical for all license plates. The Arabic numbers in the plate number region need to be segmented for the recognition task. A bunch of image processing routines such as grayscale transformation, Gaussian noise removal, thresholding, morphological operations, and a sliding bounding box is employed to handle that task. Initially, the color image is converted to the grayscale image, and a noise removal procedure is applied to the grayscale images. The Otsu method-based foreground segmentation is utilized for binarization of the denoised images where the Arabic characters are indicated with logic 1 (white color). Morphological operations such as erosion and dilation are used to remove unwanted binary artifacts [2]. After morphological operations, a sliding bounding box is used to detect each of the plate numbers. The bounding box is a single rectangular box that shows where each plate number is located [2, 21]. The segmented numbers are saved for deep feature extraction. The city regions extracted after the SegNet segmentation are also saved for deep feature extraction.

Before deep feature extraction, all Arabic numbers and the city regions are resized to 224×224 to be compatible with the input of the ResNet-18 model. The ResNet-18 architecture is a deep CNN architecture with 18 layers. The majority of these layers are convolutional, with a few pooling layers. The ResNet-18 model's fully connected layer (fc1000) is used to obtain the deep features for the cropped Arabic numbers and

city regions, respectively. The extracted features are 1000 dimensional, and these features are fed into the sequential input layer of the Long Short-Term Memory (LSTM) classifier. The bidirectional LSTM (biLSTM) layer comes after the sequential input layer. Fully connected, Softmax and classification layers follow each other to detect the Arabic numbers and cities.

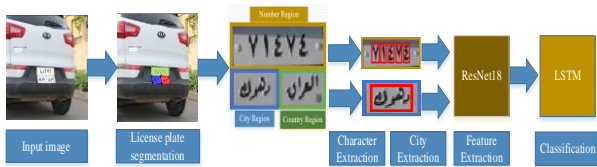


Figure 1. The proposed approach for license plate recognition of the northern Iraq vehicles

2.1 Convolutional Neural Networks (CNN)

A CNN model is made up of several layers that are used to construct different structures that lead to various tasks. Convolution, pooling, normalization, and fully connected layers are among these layers [22]. By using some $n \times n$ weight matrixes called filters, the convolution layers are responsible for feature extraction. Eq. (1) shows the output of the obtained feature maps where X_i^{l-1} X the features that are extracted from the previous layers are. Moreover, b_j^l shows the bias and w_j^l indicate the weights of the filters.

$$X_j^l = f \left(\sum_{i \in M_j} X_i^{l-1} * w_{ij}^l \right) \quad (1)$$

Besides, M_j shows the input map selection, and $f(\cdot)$ indicates the activation function, respectively. The feature maps distributed from the convolution layer are downsampled using the pooling layer. Various pooling techniques, such as average and optimal pooling, have been used in the literature. Pooling layers limit the number of computing nodes, and they avoid overfitting in CNN architecture [4]. The following is a list of the pooling:

$$X_j^l = \text{down}(X_j^{l-1}) \quad (2)$$

where the $\text{down}(\cdot)$ shows the downsampling. For the following layers, the downsampling describes local characteristics. Layers that are fully connected (FC) transfer all connections and activations from the previous layer. The FC layer provides distinguishing properties that allow the input image to be classified into different groups. The FC layer's activations are measured using matrix multiplication and a bias offset [22]. The CNN is trained using an optimization algorithm described in Eq. (3). Adaptive moment estimation (ADAM) and stochastic gradient descent with momentum (SGDM) are two well-known training approaches for neural networks. The SGDM method's weights are updated regularly for each training set to reach the goal as soon as possible [23].

$$V_t = \beta V_{t-1} + \alpha \nabla_w L(W, X, y) \quad (3)$$

W , L , and W , respectively, denote the weights, learning rate, and loss function. The new weights are calculated as follows using CNN training:

$$W = W - \alpha V_t \quad (4)$$

In the RMSProp method [24], the optimizer of ADAM uses the mean of the second moments of slopes to change the learning rate in each iteration and follow the learning rate parameter based on the mean of the first moment.

2.2 The residual neural networks (ResNet)

He et al. created the Residual neural networks (ResNet) with a 152-layer deep CNN architecture [24]. The ResNet tries to solve the vanishing gradients problem that occurs during CNN back-propagation. The ResNet architecture presented residual connections (skip connections) to avoid information loss during deep network training. The skip connection technique allows very deep networks to be trained to increase model efficiency. The ResNet architecture's core building blocks are residual. The consecutive hidden layers are connected in shallow neural networks, while ResNet's architecture includes connections through residual blocks. Two of the essential benefits of residual connections in the ResNet architecture are the retention of acquired experience during training sessions and the increase in network bandwidth, resulting in reduced model training time. The ResNet-50, a residual deep learning network with 50 layers, was the subject of this research. The ResNet-18 architecture is a deep CNN architecture with 18 layers.

2.3 LSTM

The LSTM is a kind of recurrent neural network (RNN) unique [25-27]. Because of memory blocks and recurrent links in the recurrent hidden layer, it is more stable than feed-forward neural networks and RNNs [28, 29]. In classification and regression problems, the LSTM is very successful [25, 26]. The LSTM memory blocks have self-connected memory cells that store the network's transient states at each time level. The gates provide information flow from the data to the memory cells and the other devices. The forget gate is used to scale the cell's internal state until it is inserted into the memory cell. That is accomplished by repeating the memory cell and, if possible, resetting or forgetting the cell's memory. The forget gate is regulated by a one-layer neural network with the following activation function:

$$f_t = \alpha(W[x_t, h_{t-1}, C_{t-1}] + b_f) \quad (5)$$

where, $C(t-1)$, $h(t-1)$, x_t , and b_f are the previous LSTM block memory, the output of the last block, the input sequence, and the bias vector, respectively. The weight vector assigned to each input and the logistic sigmoid function is denoted by α and W , respectively. The previous memory block has the activation function implemented. With element-wise multiplication, it calculates the effect of previous memory blocks on the ongoing LSTM. The activation vector output value is tested, and if it is close to zero, the previous memory is erased. A primary neural network creates the current memory in the input gate by taking into account the effect of the previous memory block and the tanh activation mechanism. The following is the procedure:

$$i_t = \alpha(W[x_t, h_{t-1}, C_{t-1}] + b_i) \quad (6)$$

$$C_t = f_t C_{t-1} + i_t \tanh(W[x_t, h_{t-1}, C_{t-1}] + b_c) \quad (7)$$

where it , bi , and W indicate the output of the input gate, the bias vector, and the weights, respectively. $h(t-1)$ denotes the output of the previous block, $C(t-1)$ shows the previous LSTM memory, and the α parameter indicates the activation function [25, 26, 28]. In terms of the output gate, it's a branch where the current LSTM block's output is provided using the following equations:

$$o_t = \alpha(W[x_t, h_{t-1}, C_{t-1}] + b_o) \tag{8}$$

$$h_t = \tanh(C_t) \cdot o_t \tag{9}$$

3. EXPERIMENTAL WORKS AND RESULTS

The experimental works were carried out on MATLAB with a workstation having an M4000 GPU. The vehicle images in the dataset were captured in real-time using two handheld

digital cameras from various angles and positions. The images were taken with a Canon 60D camera and an EFS 18-55mm lens, as well as a Nikon DX camera and an AF-S NIKKOR 18-105mm lens. 4288×2848 and 5184×3456 were the sensor resolution settings, respectively. Six hundred vehicle images were taken in various lighting conditions and different weather conditions, including heat, wind, snow, rain, and dust cloud, respectively. Images of various vehicle types, including cars, vans, buses, and minibusses, are included in the latest dataset. Figure 2 displays a selection of vehicle images from the dataset, including a variety of vehicle models and photos taken under various weather conditions.

Vehicle license plates in northern Iraq come in a variety of colors. They are three distinct regions: a plate number area, a city region, and a country region, as previously stated. Figure 3b shows several examples of license plates. The number region is the upper region of the license plate that shows the plate number, as seen in Figure 3a.

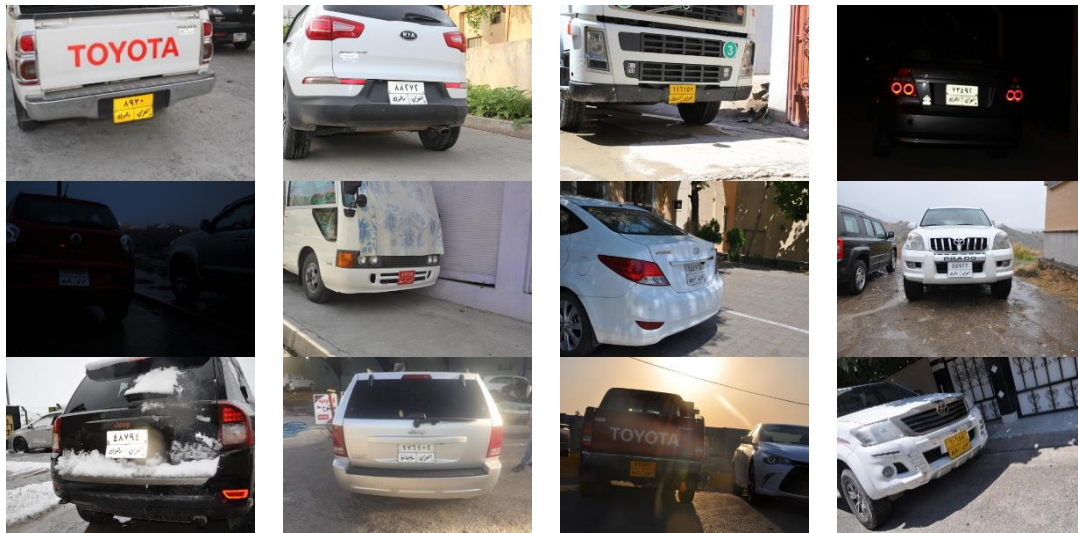


Figure 2. The captured images for the dataset

Since the national regions were similar, it was eliminated. The plate number and the name of the city are written in Arabic. The length of the plate number is not identical. Figure 4 depicts the relationship between Arabic numerals and their Latin equivalents.



Figure 4. Arabic numbers (top row) and Latin numbers (bottom row)

Erbil, Duhok, and Sulemania are the three cities that make up northern Iraq. Figure 5 shows a license plate illustration of these three towns.

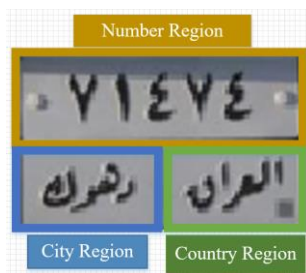


Figure 3a. The structure of the license plates in northern Iraq



Figure 5. Northern Iraq city names on license plates



A randomly selected 75% of the dataset was used for training purposes, with the remaining 25% used for testing the proposed method. The SegNet architecture, which was previously trained [2], was used for the segmentation of the license plates. After segmentation of the region of interest in the license plate regions, a character segmentation procedure was employed for Arabic number extraction. Besides, the segmented city regions were directly used in ResNet18-based

Figure 3b. Some sample license plates from northern Iraq

deep feature extraction. Both extracted Arabic numbers and the city regions were resized to 224×224, and the 'fc1000' layer of the ResNet18 architecture was used for 1000-dimensional deep feature extraction. After feature extraction, the LSTM was used for classification. The training of the LSTM was carried out using an "ADAM" optimizer with a batch size of 25 and an initial learning rate of 0.0001. The learning rate was dropped once after every 1000 iterations by a factor of 100, and the optimizer trained the network for 30 epochs. The training progress of the LSTM network was given in Figure 6. As shown in Figure 6, the training was completed after 4500 iterations, taking 2 minutes and 15 seconds. These parameters were determined by the trial and error method. The classification accuracy was 98.51%. The loss value was over 2 at the beginning of the training of the LSTM network. At the 500th iteration, the loss value came below 0.5, and at the end of the training; it was quite close to 0.

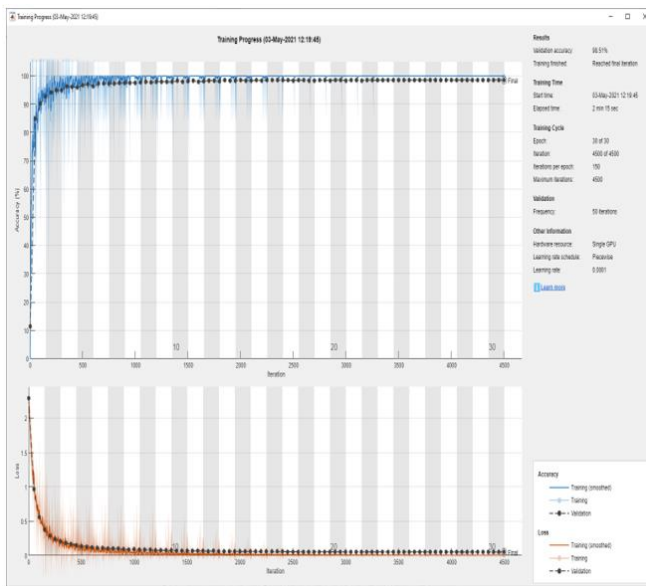


Figure 6. The training progress of the LSTM for Arabic numbers

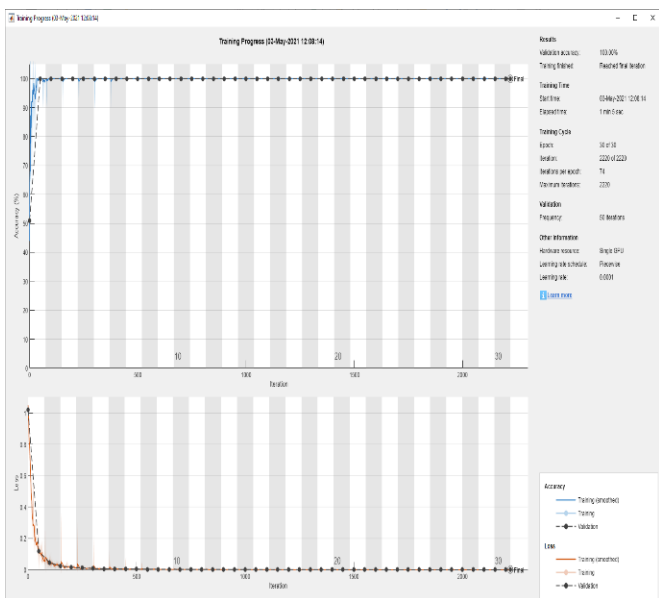


Figure 7. The training progress of the LSTM for city classification
In Figure 7, the training progress for city classification was

given. As seen in Figure 7, the training and test accuracy scores were around 50% at the beginning of the training, and the accuracy scores for the training and test dataset were increased to 100% abruptly around the 70th iteration.

Table 1 shows the confusion matrix for Arabic number classification. While the rows show the output classes, the columns show the target classes. When the confusion matrix was examined based on the output classes, it was seen that numbers 5, 6, 7, and 8 were classified with 100% accuracy scores. 96.2% accuracy score was obtained for number 2 which was the worst among all. Besides, when the results were evaluated based on the target classes, it was seen that only number 0 was classified with a 100% accuracy score. In addition, above 98% accuracy scores were obtained for numbers 1, 2, 3, 4, 6, 8, and 9, respectively. Number 5 was classified with a 97.2% accuracy score, which was the worst one in the target classes.

Table 1. Testing accuracy confusion matrix for Arabic number classification

Output Class	0	1	2	3	4	5	6	7	8	9	Accuracy
0	69 8.6%	1 0.1%	0 0.0%	0 0.0%	1 0.1%	1 0.1%	0 0.0%	1 0.1%	0 0.0%	1 0.1%	93.2%
1	0 0.0%	118 14.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.1%	0 0.0%	0 0.0%	99.2%
2	0 0.0%	1 0.1%	75 9.3%	0 0.0%	0 0.0%	1 0.1%	0 0.0%	0 0.0%	1 0.1%	0 0.0%	96.2%
3	0 0.0%	0 0.0%	0 0.0%	93 11.6%	0 0.0%	0 0.0%	1 0.1%	0 0.0%	0 0.0%	0 0.0%	98.9%
4	0 0.0%	0 0.0%	0 0.0%	1 0.1%	72 9.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	98.6%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	70 8.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	71 8.8%	0 0.0%	0 0.0%	0 0.0%	100%
7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	72 9.0%	0 0.0%	0 0.0%	100%
8	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	67 8.3%	0 0.0%	100%
9	0 0.0%	0 0.0%	1 0.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	85 10.6%	98.8%
	100%	98.3%	98.7%	98.9%	98.6%	97.2%	98.6%	97.3%	98.5%	98.8%	98.5%
	0.0%	1.7%	1.3%	1.1%	1.4%	2.8%	1.4%	2.7%	1.5%	1.2%	1.5%

Later, these obtained 100% accuracy scores were continued throughout the training progress for both training and testing datasets. The training was completed after 2200 iterations, which took 1 minute and 5 seconds. Similarly, the loss curve for city classification started around 1 and dropped to zero around the 500th iteration. t kept going on zero until the end of its training progression.



Figure 8. Successfully detected/recognized license plates

Table 2 shows the confusion matrix for testing the accuracy of the city classification. As it was mentioned earlier, the rows and columns of Table 2 show the classification accuracy scores for the output classes and target classes, respectively. As the training and testing were completed with 100% accuracy scores, the obtained class-based accuracy scores were also 100% for all city categories.

Table 2. Testing accuracy confusion matrix for city classification

Output Class	Duhok	159 40.1%	0 0.0%	0 0.0%	100% 0.0%
	Erbil	0 0.0%	112 28.2%	0 0.0%	100% 0.0%
	Sulemania	0 0.0%	0 0.0%	126 31.7%	100% 0.0%
		100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%
	Target Class	Duhok	Erbil	Sulemania	

Figure 8 indicates some cases of license plates that were successfully identified and classified. In the given images, the corresponding Latin numbers for the recognized Arabic numerals appear above the numbers on the plate, and the recognized city labels appear underneath the city area. Figure 8 shows that three Duhok city labels and one Erbil city label were correctly identified.

Table 3 shows the results of a further comparison of Arabic plate number identification. In all compared methods, the authors used deep learning-based approaches in their proposed methods. In terms of accuracy scores, the proposed method outperformed Shivakumara et al. [11] and Cheang et al. [8] methods, where 97.15% and 98.15% accuracy scores were reported. But the method proposed by Omar et al. [2] produced a better accuracy score than our proposed method.

Table 3. Performance comparison of the proposed method and existing methods (Arabic number recognition)

Method	Accuracy
Shivakumara et al. [11]	97.15%
Cheang et al. [8]	98.15%
Omar et al. [2]	99.37%
Proposed method	98.51%

4. CONCLUSIONS

In this paper, an approach was proposed for LP detection and recognition. The proposed method achieved accuracy scores were 98.51% and 100% for Arabic number and city marking classifications. As we proposed an efficient scheme, the proposed method has several limitations. Since the

semantic segmentation was at the center of the proposed work, its outputs had a significant impact on the final identification. The Arabic numbers could not be correctly segmented, and character segmentation was also an essential part of the plate number identification process. As a result, we want to boost semantic segmentation in the future by increasing the number of images in the dataset. We intend to suggest a novel approach that will eliminate the need for a segmentation point, as character segmentation has been missing in some cases. We will also collect more vehicle images for improving the LP detection for Arabic numbers and alphabets.

REFERENCES

- [1] Omar, N., Abdulazeez, A.M., Sengur, A., Al-Ali, S.G.S. (2020). Fused faster RCNNs for efficient detection of the license plates. *Indonesian Journal of Electrical Engineering and Computer Science*, 19(2): 974-982. <https://doi.org/10.11591/ijeecs.v19.i2.pp974-982>
- [2] Omar, N., Sengur, A., Al-Ali, S.G.S. (2020). Cascaded deep learning-based efficient approach for license plate detection and recognition. *Expert Systems with Applications*, 149: 113280. <https://doi.org/10.1016/j.eswa.2020.113280>
- [3] Kakani, B.V., Gandhi, D., Jani, S. (2017). Improved OCR based automatic vehicle number plate recognition using features trained neural network. In 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp. 1-6. <https://doi.org/10.1109/ICCCNT.2017.8203916>
- [4] Prabhakar, P., Anupama, P., Resmi, S.R. (2014). Automatic vehicle number plate detection and recognition. In 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), pp. 185-190. <https://doi.org/10.1109/ICCICCT.2014.6992954>
- [5] Singh, B., Kaur, M., Singh, D., Singh, G. (2016). Automatic number plate recognition system by character position method. *International Journal of Vision Robotics*, 6(1/2): 94-112.
- [6] Li, H., Wang, P., Shen, C. (2018). Toward end-to-end car license plate detection and recognition with deep neural networks. *IEEE Transactions on Intelligent Transportation Systems*, 20(3): 1126-1136. <https://doi.org/10.1109/TITS.2018.2847291>
- [7] Montazzolli, S., Jung, C. (2017). Real-time Brazilian license plate detection and recognition using deep convolutional neural networks. In 2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), pp. 55-62. IEEE. <https://doi.org/10.1109/SIBGRAPI.2017.14>
- [8] Cheang, T.K., Chong, Y.S., Tay, Y.H. (2017). Segmentation-free vehicle license plate recognition using ConvNet-RNN. *arXiv preprint arXiv:1701.06439*.
- [9] Zang, D., Chai, Z., Zhang, J., Zhang, D., Cheng, J. (2015). Vehicle license plate recognition using visual attention model and deep learning. *Journal of Electronic Imaging*, 24(3): 033001-033001. <https://doi.org/10.1117/1.JEI.24.3.033001>
- [10] Bulan, O., Kozitsky, V., Ramesh, P., Shreve, M. (2017). Segmentation-and annotation-free license plate recognition with deep localization and failure

- identification. *IEEE Transactions on Intelligent Transportation Systems*, 18(9): 2351-2363. <https://doi.org/10.1109/TITS.2016.2639020>
- [11] Shivakumara, P., Tang, D., Asadzadehkaljahi, M., Lu, T., Pal, U., Hossein Anisi, M. (2018). CNN-RNN based method for license plate recognition. *CAAI Transactions on Intelligence Technology*, 3(3): 169-175. <https://doi.org/10.1049/trit.2018.1015>
- [12] Yang, Y., Li, D., Duan, Z. (2018). Chinese vehicle license plate recognition using kernel-based extreme learning machine with deep convolutional features. *IET Intelligent Transport Systems*, 12(3): 213-219. <https://doi.org/10.1049/iet-its.2017.0136>
- [13] Kim, S.G., Jeon, H.G., Koo, H.I. (2017). Deep-learning-based license plate detection method using vehicle region extraction. *Electronics Letters*, 53(15): 1034-1036. <https://doi.org/10.1049/el.2017.1373>
- [14] Selmi, Z., Halima, M.B., Alimi, A.M. (2017). Deep learning system for automatic license plate detection and recognition. In 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), 1: 1132-1138. <https://doi.org/10.1109/ICDAR.2017.187>
- [15] Yaseen, N.O., Al-Ali, S.G.S., Sengur, A. (2019). An efficient model for automatic number plate detection using HOG feature from new north Iraq vehicle images dataset. In 2019 1st International Informatics and Software Engineering Conference (UBMYK), pp. 1-6. <https://doi.org/10.1109/UBMYK48245.2019.8965573>
- [16] Wang, W., Yang, J., Chen, M., Wang, P. (2019). A light CNN for end-to-end car license plates detection and recognition. *IEEE Access*, 7: 173875-173883. <https://doi.org/10.1109/ACCESS.2019.2956357>
- [17] Henry, C., Ahn, S.Y., Lee, S.W. (2020). Multinational license plate recognition using generalized character sequence detection. *IEEE Access*, 8: 35185-35199. <https://doi.org/10.1109/ACCESS.2020.2974973>
- [18] Izidio, D.M., Ferreira, A., Medeiros, H.R., Barros, E.N. D.S. (2020). An embedded automatic license plate recognition system using deep learning. *Design Automation for Embedded Systems*, 24(1): 23-43. <https://doi.org/10.1007/s10617-019-09230-5>
- [19] Dong, M., He, D., Luo, C., Liu, D., Zeng, W. (2017). A CNN-based approach for automatic license plate recognition in the wild. In *BMVC*.
- [20] Kendall, A., Badrinarayanan, V., Cipolla, R. (2015). Bayesian SegNet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. arXiv preprint arXiv:1511.02680.
- [21] Choeychuen, K., Kumhom, P., Chamnongthai, K. (2005). An efficient implementation of the nearest neighbor based visual objects tracking. In 2006 International Symposium on Intelligent Signal Processing and Communications, pp. 574-577. <https://doi.org/10.1109/ISPACS.2006.364723>
- [22] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-9. <https://doi.org/10.1109/CVPR.2015.7298594>
- [23] Asif, M.R., Chun, Q., Hussain, S., Fareed, M.S. (2016). Multiple license plate detection for Chinese vehicles in dense traffic scenarios. *IET Intelligent Transport Systems*, 10(8): 535-544. <https://doi.org/10.1049/iet-its.2016.0008>
- [24] He, T., Yao, J., Zhang, K., Hou, Y., Han, S. (2014). Accurate multi-scale license plate localization via image saliency. In 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), pp. 1567-1572. <https://doi.org/10.1109/ITSC.2014.6957916>
- [25] Hochreiter, S., Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8): 735-1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- [26] Sajjad, M., Kwon, S. (2020). Clustering-based speech emotion recognition by incorporating learned features and deep BiLSTM. *IEEE Access*, 8: 9861-79875. <https://doi.org/10.1109/ACCESS.2020.2990405>
- [27] Deniz, E., Akpınar, M.H., Sengur, A. (2022). Bidirectional LSTM based harmonic prediction. *VI-International European Conference on Interdisciplinary Scientific Research*, August 26-27, 2022, Bucharest, Romania, pp. 70-79.
- [28] Demir, F., Sobahi, N., Siuly, S., Sengur, A. (2021). Exploring deep learning features for automatic classification of human emotion using EEG rhythms. *IEEE Sensors Journal*, 21(13): 4923-14930. <https://doi.org/10.1109/JSEN.2021.3070373>
- [29] Jiang, X.P., Wang, Q., Long, Y.K., Zhang, S.L., Fang, Y., Hu, D. (2022). Vibration signal features prediction of GIS equipment based on improved slime mold optimization algorithm optimizing CNN-BiLSTM. *Traitement du Signal*, 39(4): 1109-1117. <https://doi.org/10.18280/ts.390403>