Turkish Vehicle License Plate Recognition Using Deep Learning

Irfan Kilic, Galip Aydin
Department of Computer Engineering
Firat University
Elazig, Turkey
{irfankilic, gaydin}@firat.edu.tr

***Abstract*— License plate recognition systems are commonly used in various real-life applications. Traditionally, several image processing techniques have been used in these systems. Although these solutions provide adequate results in general, in some cases incorrect results are obtained. In this study we use Deep learning to provide a highly accurate license plate recognition system. We also employ image processing steps in the process to increase the accuracy of the results from convolutional neural. The system uses TensorFlow framework in the background and developed using Keras deep learning library. We created a new Turkish vehicle license plate dataset for this study using real world vehicle images obtained from Firat University security cameras. The results show that the system performs well and the trained neural network can highly accurately recognizes the license plates.**

***Index Terms*— License plate recognition, Deep learning, TensorFlow, Keras, Convolutional neural networks**

I. INTRODUCTION

Automatic License Plate Recognition Systems (ALPR) are one of the most frequently used image processing systems. They are used in school campuses, public car parks or in urban traffic to recognize license plates for security purposes. In existing ALPR systems, camera images and RFID card solutions are used separately or together [1,2]. However, these systems still do not produce the desired high accuracy. Especially when the city traffic is considered, the performance of these systems in real time is far away from the desired level. It is critical in today's conditions that these systems give correct results with near-zero errors. Today, together with the recent advances in both hardware and software systems, a large amount of data are being collected with the widespread use of these technologies. By using these data, the bottlenecks in the existing ALPR systems can be overcome with the help of deep learning algorithms which use deep neural networks..

License plate detection is the process of determining the location of the license plate area on the image which can be performed using image processing or deep learning methods. At the end of this process the coordinates of the vehicle license plate are estimated.

978-1-5386-6878-8/18/$31.00 ©2018 IEEE

These coordinates can be expressed as (x_1, y_1), (x_2, y_2). In addition, the width (g) and height (y) information can be obtained by this process. Fig. 1 shows the location of the plate on a real-world image from the dataset. License plate recognition is the process of recognizing characters and digits with optical character recognition (OCR) or deep learning techniques. Fig. 1 shows the result of the plate area detection and Fig 2 shows the recognition results on a vehicle image. At the end of these two steps, the ALPR systems are expected to provide a character sequence consisting of at least 6 or 8 characters

Fig. 1. License plate location on sample image

Fig. 2. License plate recognition and expected result

Several studies have been conducted in this area. Yadnesh Joshi et al. [2] proposed smart park management using OCR and RFID. In the proposed system with RFID and OCR enabled, an automated system for parking management is provided. Various techniques have been used for the detection of plate from raw images in the context of plate detection/recognition. One of these is the SCW (Sliding Concentric Window), a two-step method with two concentric windows moving from the upper left corner of the view [3,4]. Another is to use a median filter, especially to reduce noise in the image [5]. The SURF technique applied to the Hessian matrix is used for license plate detection [6]. The Ransac technique has also been used to detect impairments in the visual [6]. Another technique based on rectangular features and multiple thresholds is also proposed by Ihsan Ullah [7]. The local object appearance in an image for object detection and the distribution of density gradients or edge directions of the shape can be defined by HOG identifiers [8,9]. Edge detection algorithms such as Canny, Canny-Deriche, Differential, Sobel, Prewitt and Roberts Cross have also been used for plate detection [3,6,10]. Smearing algorithm has been used for plate detection because of its simple and fast solution [11].

In his Master's thesis [12] Naga Surya Sandeep Angara, explains how to implement an ALPR system using deep learning techniques. In a study by Syafeeza Ahmad Radzi and Mohamed Khalil-Hani [13] recognition of the characters of vehicle license plates was carried out using CNN. A study by Syed Zain Masood et al. [14] designed an ALPR system. The system's core technology was built using a series of deep Convolutional Neural Networks (CNN) combined with accurate and efficient algorithms [14]. A study by Christian Gerber and Mokdong Chung [15] proposed a method for obtaining improved plate detection by applying multiple convolutional neural networks (CNN) for mobile devices.

In the study by Meng Dong et al. [16], the automatic plate recognition (ALPR) system was tested in natural environments. Such an ALPR system takes ordinary images as input and extracts recognized plate numbers. In the detection phase, a hierarchical structure including a rapid regional proposal network and an R-CNN network has been used. A study by Francisco Delmar Kurpiel et al. [17] proposed an approach that models a function based on a convolutional neural network which produces a score for each image sub-region for detecting the license plates. A new convolutional neural network (CNN) based method has been proposed by Lele Xie et al. [18] for high-precision real-time vehicle license plate recognition. In the proposed method, CNN based MD-YOLO framework is used for multi-directional vehicle license plate detection.

II. DATASET PREPARATION

In this section, we describe the new license plate dataset which has been prepared by using the vehicle images with Turkish license plates. The most appropriate model for license plate recognition with deep learning is the supervised learning model. Therefore, tagged data must be prepared. The aim of this study is to prepare an original data set specific to Turkish vehicle license plates and to use them to train a neural network. Smearing algorithm [11] was used in the first step for the dataset preparation and tagging was done manually.

A. Data collection

Vehicle images were taken from IP security cameras installed at the Firat University vehicle gates. No preprocessing such as proximity or distance detection, brightness and transparency adjustment on the obtained images have not been performed. The original images have resolution of 640x360 pixels. A total of 34,580 images were were. A sample image is given in Fig. 1.

B. Plate detection with smearing algorithm

The smearing algorithm is applied to binary images for plate detection [11]. For the correct operation of the algorithm, it is very important to extract the image from the foreground [11]. Our goal here is to create a dataset for deep learning network without performing any cleaning on the data, so a direct smearing algorithm has been applied. The algorithm scans images vertically and horizontally [11]. The number of white pixels obtained from these scans is critical [11]. If the number of white pixels is less than or equal to the specified threshold, this pixel is converted to a black pixel [11]. A Matlab program is developed for the implementation of the algorithm. Application steps are as follows: In the first step, images are loaded and cropped to improve the performance and accuracy of the algorithm. In the second step, the image is converted to grayscale and then converted into a binary image using the Otsu method, which sets the threshold value according to the image properties. In the third step, morphology was performed with a filter (kernel) in the form of a 5x5 rectangle to reduce noise on the image [19]. Fig. 3 shows the resulting image after this process is completed. Smearing algorithm was applied twice in the fourth and fifth steps.

Fig. 3. Appearance after Otsu and Morphology

C. Data tagging

At this stage, about 30% of the 34,580 images were cropped to show the vehicle licence plate and its surroundings. After removing the duplicate plates from the dataset 10,374 cropped images remained. Manual labeling was performed on these images. Each cropped image is named with the vehicle license plate number. For example, the file "1 (23) .jpg" is named "99ABC123.jpg". A total of 4693 plates were manually labeled.

D. Data structure and preparation

After labeling is done, the images are divided into 3 groups: 75% for training, 5% for validation, 20% for testing. The "json" files, which hold various information for each license plate with the same name as the image are created. TABLE I shows a sample json file with aforementioned contents.

TABLE I. "CONTENTS OF "JSON" FILE


```
{"description": "2*****13", "tags": [], "objects": [],
"size": {"width": 100, "height": 26}, "name":
"2*****13"}
```


The folder structure for the prepared dataset is given in Fig. 4.

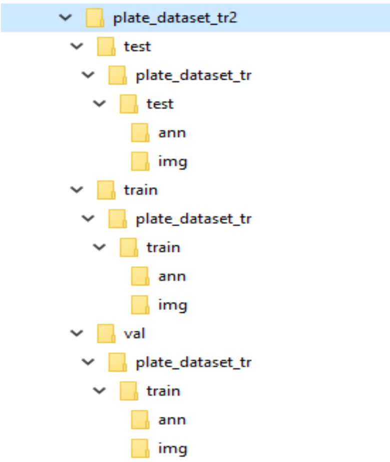

Fig. 4. Folder structure of the dataset

III. IMPLEMENTATION AND RESULTS

We have created a deep neural network model using Keras for the dataset we prepared for license plate recognition. The created model was trained on the training dataset and tested on the test dataset. In this way, an OCR-like application has been developed using deep learning methods.

A. Implementation details

We used Python 3.5 [20] for the system development. The application was run on a server with Nvidia 8 GB GTX 1080 GPU card on Ubuntu 16.04. Since the images in our dataset are real-world images, some pre-processing is required on the plate images using OpenCV. The resolutions of the images are 100x26 pixels. Image preprocessing steps are given in Fig. 5.

As shown in Fig. 5, the Median Blur smoothing [21] is applied to the gray-channel image using the OpenCV library, first of all taking the median of all pixels under the filter area and changing the central element with this median value. Adaptive Gaussian thresholding [22] was then applied to calculate the threshold for small regions of the image and to provide better results for images with different lighting characteristics. Finally, Morphological Transformation [19], which is useful for closing small holes or small black dots on objects in the foreground, has been applied. Fig. 6 shows a pre-processed license plate image.

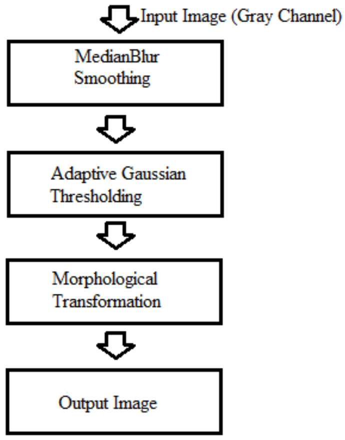

Fig. 5. Image preprocessing steps

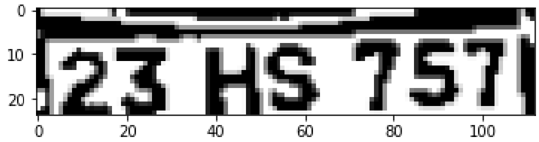

Fig. 6. License plate image after preprocessing

B. Creating training model

After the preparations were completed, we build the deep neural network model. The model used in this study is based on [23] with several modifications. Our model is given in Fig. 7. In this model, CNN is first applied to the pre-processed input image, then the image properties obtained from CNN are given to the LSTM network. Finally, the decryption algorithm is applied to the output of the LSTM network to achieve the desired result.

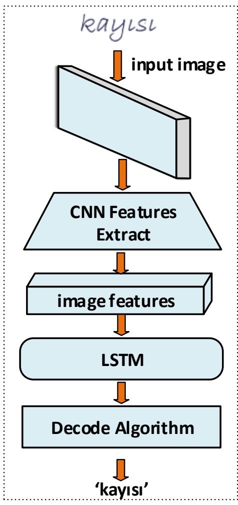

Fig. 7. Our training model overview

We send the image tensor to the CNN feature extractor and we produce the tensor in the form of 4x8x4. To understand this, we leave the "kayısı" image to the property tensor. In this case, tensor height is 4, width is 8 and the channel number is 4. Thus, a 3-channel input image is converted into a 4-channel image. Then we reshape the output (16x8). After that, we obtain a vector of 16 with 8 elements. Consecutively, these 8 vectors are sent to LSTM network and output is obtained. The softmax layer (SM) is followed by a fully connected (FC) layer. As a result, 8 vectors of 6 elements are obtained. Each of these vectors contain the probability distribution of the alphabet symbols at each LSTM step. At each step we get the most probable symbol and in turn obtain the eight-character sequence. Then, all consecutive repeating characters are reduced to a character. The special space character allows us to split repeated symbols in the original label. An empty symbol has been added to the alphabet to teach the neural network to find the space between such symbols. Then all empty symbols are removed. While the network is trained, the decoding algorithm is changed with the CTC Loss layer. The layers of the CNN are shown in Fig. 8.

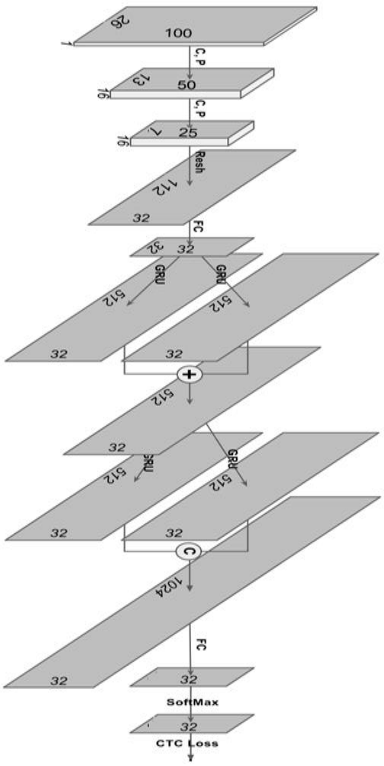

Fig. 8. Deep neural network model [25]

C. Results

The model was applied to the images in the test set after being trained on the training set and very high accuracy was obtained. Also the probability distribution of each step of the RNN were visualized as a matrix. Fig. 9 shows the results and probability distribution matrix generated by our network on a test image.

Fig. 9. Test image and probability distribution matrix

The results we obtain by using this deep neural network architecture on the aforementioned dataset are given in TABLE II and TABLE III.

TABLE II. Loss and accuracy values

	Loss	Accuracy (%)
Test images	1.0833	88.76
Validation images	1.1562	90.21

TABLE III. Plate and character based test results

	Accuracy (%)
By plate	96.36
By number	99.43
By letter	99.05
By all characters	99.31

IV. Conclusions

In this study, a deep learning model was created, trained and tested on the dataset we prepared for license plate recognition. The training results show that although loss and accuracy results are good, they are not at the desired level. When the model is evaluated on a license plate basis, very good results are obtained. We expect the results to improve with the increase in the number of images in our dataset.

References

[1] M. Mohandes, M. Deriche, H. Ahmadi, M. Kousa, and A. Balghonaim, “An intelligent system for vehicle access control using RFID and ALPR technologies,” Arabian Journal for Science and Engineering, vol. 41(9), pp. 3521-3530, 2016.

[2] Y. Joshi, P. Gharate, C. Ahire, N. Alai, and S. Sonavane, “Smart parking management system using RFID and OCR,” International Conference on Energy Systems and Applications, 2015, pp. 729-734.

[3] C. Patel D. Shah and A. Patel, “Automatic Number Plate Recognition System (ANPR): A Survey,” International Journal of Computer Applications, vol. 69(9), 2013.

[4] C.N.E. Anagnostopoulos, I.E. Anagnostopoulos, V. Loumos, and E. Kayafas, “A license plate-recognition algorithm for intelligent transportation system applications,” IEEE Transactions on Intelligent transportation systems, vol. 7(3), pp. 377-392, 2006.

[5] C. Oz and R. Köker, “Vehicle Licence Plate Recognition Using Artificial Neural Networks,” ELECO’2001 2nd Intern. Conf. on Electrical and Electronics Engineering, Bursa, 2001.

[6] N. Gupta T. Sandep P. Gupta D. Goyal and M. Goyal, “A Review: Recognition of Automatic License Plate in Image Processing,” Advances in Computational Sciences and Technology ISSN 0973-6107, vol. 10(5), pp. 771-779, 2017.

[7] I. Ullah, and H.J. Lee, “License Plate Detection Based on Rectangular Features and Multilevel Thresholding,” In Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV) (p. 153). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing, 2016.

[8] A. Sharma, A. Kumar, K.V.S. Raja and S. Ladha “Automatic License Plate Detection,” CS771 Course Project, Winter Semester 2015-16, Indian Institute of Technology Kanpur, 2015.

[9] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 886-893, 2005.

[10] R. Collobert, and J. Weston, “A unified architecture for natural language processing: deep neural networks with multitask learning,” In Proceedings of the 25th international conference on Machine learning (ICML '08). ACM, New York, NY, USA, pp. 160-167, 2008.

[11] M. Çiçek, “Araç Plakası Tanınması,” Yüksek Lisans Tezi, Haliç Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği, Mayıs 2008.

[12] N.S.S. Angara, “Automatic License Plate Recognition Using Deep Learning Techniques,” Electrical Engineering Theses. 2015.

[13] S.A. Radzi and M. Khalil-Hani, “Character recognition of license plate number using convolutional neural network,” In International Visual Informatics Conference, pp. 45-55, 2011.

[14] S.Z. Masood, G. Shu, A. Dehghan, and E.G. Ortiz, “License Plate Detection and Recognition Using Deeply Learned Convolutional Neural Networks,” arXiv preprint arXiv:1703.07330, 2017.

[15] C. Gerber and M. Chung, “Number Plate Detection with a Multi-Convolutional Neural Network Approach with Optical Character Recognition for Mobile Devices,” Journal of Information Processing Systems, vol. 12(1), 2016.

[16] M. Dong, D. He C. Luo, D. Liu W. Zeng, “A CNN-Based Approach for Automatic License Plate Recognition in the Wild,” British Machine Vision Conference, 2017.

[17] F.D. Kurpiel, R. Minetto and B.T. Nassu, “Convolutional neural networks for license plate detection in images,” International Conference on Image Processing (ICIP), pp. 3395-3399, 2017.

[18] L. Xie, T. Ahmad, L. Jin, Y. Liu and S. Zhang, “A New CNN-Based Method for Multi-Directional Car License Plate Detection,” IEEE Transactions on Intelligent Transportation Systems, 2018.

[19] R. van den Boomgard and R van Balen R., “Methods for Fast Morphological Image Transforms Using Bitmapped Images,” Computer Vision, Graphics, and Image Processing: Graphical Models and Image Processing, vol. 54(3), pp. 252–254, 1992.

[20] https://www.python.org/, Retrieved on June 4, 2018.

[21] https://docs.opencv.org/3.1.0/d4/d13/tutorial_py_filtering.html, /, Retrieved on June 4, 2018.

[22] https://docs.opencv.org/3.4.0/d7/d4d/tutorial_py_thresholding.html, Retrieved on June 4, 2018.

[23] https://hackernoon.com/latest-deep-learning-ocr-with-keras-and-supervisely-in-15-minutes-34aecd630ed8, Retrieved on Feb 2, 2018.