License Plate Detection and Recognition Using Convolution Networks

Aleksandra Danilenko
Software department
Samara National Research University
Samara, Russia
danilenko.al@gmail.com

***Abstract*—The paper presents the implementation of a system for license plate detection and recognition in digital images. A comparative analysis of the detection methods was carried out. It led to the selection of the convolutional neural network for the detection module. The customized network architecture was developed on the basis of the neural network GoogleNet. An overview of approaches for license plate detection and recognition in digital images was made. The best solution for the recognition module was determined. The developed system for license plate detection and recognition can identify license plates that use letters of the Latin alphabet and Arabic digits. Testing of the detection and recognition system of the license plate was carried out. It proved the reliability of the system and its ability to perform well, even in conditions that might prevent the detection and recognition of license plates. The accuracy of the implemented system was at least 94%.**

Keywords—license plate, detection, recognition, convolution networks, comparative analysis

I. INTRODUCTION

Automatic license plate recognition is a common research topic [1,2,3] and has many practical applications, such as automatic collection of fares, enforcement of traffic rules, access control to private territories and traffic monitoring. The classical process of license plate recognition usually consists of the following steps:

- image preprocessing (smoothing, binarization, contouring);
- search and selection of a frame of license plate;
- image processing with a frame license plate (alignment of the rectangle with license plate recognition, filtering, binarization);
- image segmentation;
- classification of characters [4].

II. PROBLEM FORMULATION

The main function of the automated system is to provide user with license plate recognition results. A digital image should be input to the system. The output should be the original image with the license plate area highlighted on it and the recognition result in the form of a string value.

The subsystem should also provide the user with the opportunity to search the database of previously recognized license plates (using filters by license plates, date and time of its fixation). The search is possible only by the recognition results that are allowed according to the user's access rights after successful authorization. Accordingly, it is necessary to provide for the possibility of user authorization in the system.

The block diagram of the developed system is presented in Figure 1.

The client component includes the following components:

- an integrator with video surveillance cameras, which is necessary to obtain an image by connecting to an image source via HTTPS;
- the operator's workstation is a web-based interface for user interaction with the system.

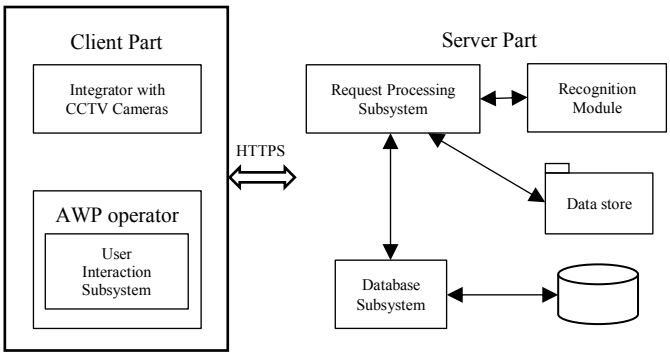

Fig. 1. System block diagram.

The server part consists of the following components:

- a request processing subsystem responsible for transmitting data from video surveillance cameras to a recognition module and subsequent processing of results, as well as user requests;
- the subsystem of working with a database, which is responsible for storing recognition results in a database and their subsequent receipt for provision at the request of users;
- recognition module, which receives an image at the input, and gives the cut-out number of the receiver and the recognition result in the form of text to the output;
- data storage, which stores the original images obtained from surveillance cameras, as well as cut out as a result of recognition of a frame license plate;
- a database of the system in which user data and recognition results are stored.

III. MATERIALS AND METHODS

A. The task of detection using convolutional neural networks

Images began to be processed by neural networks since the 1980s. The first take-off of convolutional neural networks a special architecture that is perfect for processing

precisely such inputs as digital images, dates back to the same time.

Overall, this is a rare example of an area in which neural networks have never completely disappeared from sight. However, progress in image processing also accelerated dramatically after the start of the deep learning revolution. Deep convolution networks won a series of competitions in character recognition and even recognition of video from security cameras in 2009-2010.

In addition, the first implementations of neural networks based on GPUs appeared in 2009. It gave a huge impetus to all research related to convolution networks. Convolutional neural networks are a very wide class of architectures whose main idea is to reuse the same parts of the neural network to work with different small local sections of the input image. It is worth noting that the use of neural networks allows us to better solve most of the problems of image processing [5,6,7].

B. Convolutional neural network YOLO

YOLO (You Only Look Once) is a new approach to solving the problem of detecting objects, proposed in 2015 by Joseph Redmon and other scientists [8]. This method allows us to process the image in one stage (pass), starting from the pixels of the input image and ending with the coordinates of the bounding rectangles and their probabilities.

YOLO combines the individual components of object detection into a single convolution network, which simultaneously predicts the coordinates of the bounding rectangles and the probability of the object belonging to each of the known classes. Using this approach, it is possible to train a neural network in images of any size and directly optimize detection performance. This unified model has several advantages over traditional object detection methods.

Firstly, the YOLO architecture provides real-time operation while maintaining high average accuracy. Secondly, YOLO analyzes the whole image during prediction unlike methods using sliding windows or based on local assumptions. YOLO uses the attributes of the entire image to predict each bounding box. Thus, the approach fully describes the full image of the image, as well as all objects located on it.

C. Neural network structure

To build the architecture of the neural network, the developers focused mainly on the GoogleNet model for classifying images. The YOLO configuration is implemented as a convolutional neural network, where the initial convolutional layers of the neural network extract features from the input image, and fully-connected layers predict the probabilities and coordinates of the output. The network has 24 convolutional layers followed by 2 fully connected layers. Instead of the original modules used by GoogleNet, cascades of convolutional and thinning layers are used. The output layer of the neural network is a tensor of dimension 7x7x30 [8].

Given the advantages of neural networks in the pattern recognition problem, it was decided to use convolutional neural networks in this paper to solve the problem of detecting and recognizing license plates of vehicles.

D. Proposed method of license plate detection and recognition

Training a neural network to simultaneously solve the task of license plate detection and recognition requires a huge amount of tagged data. In other words, a large training sample in which not only the areas of finding these objects will be labeled, but also symbols that should be repeated quite often, is necessary. Due to the lack of a suitable training set for solving the problems of identification, two neural networks were trained on the available datasets. The proposed method will greatly simplify the process of license plate detection and recognition, reducing it to the following steps:

- detecting and cutting the frame of license plate recognition;
- character detection;
- sorting characters.

To develop a system of license plate detection and recognition it is necessary to design and train the following machine learning models:

- detector of license plate recognition receiving the input image of the vehicle and returning the coordinates of the framing rectangle;
- a symbol detector that receives an input image of a rectangle for a symbol of resistance and returns the coordinates of the framing rectangles for symbols that leave the symbol of resistance.

The proposed pattern of recognition of license plate is presented in Figure 2.

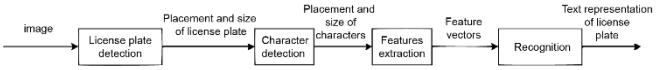

Fig. 2. The process of recognition of license plate in the image.

The system under development should be able to localize numbers that use letters of the Latin alphabet and Arabic numerals. Figure 3 shows examples of gas distribution systems in the countries of the European Union, the CIS and Russia.

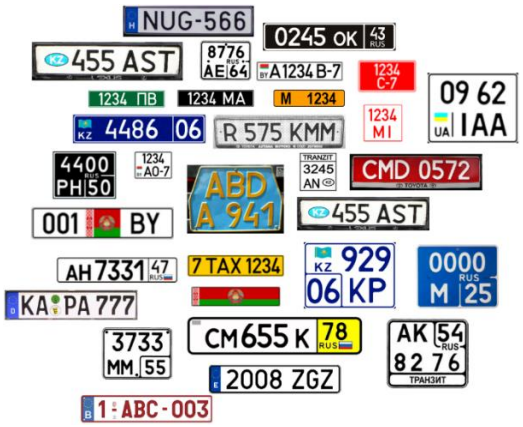

Fig. 3. Examples of license plate.

E. Selection and description of the architecture of the neural network used

An analysis of the most common neural network architectures was carried out to select the appropriate neural

network architecture for the detection of gas distribution networks. Table 1 shows a comparative analysis of several classifier architectures according to the following parameters:

- classification accuracy on the ImageNet dataset (Top-1 accuracy metric);
- number of neural network parameters, millions (Params metric);
- the number of operations of addition and multiplication, millions (metric MAdds);
- time for classification of a color image of 224 x 224 pixels in size on one core of Qualcomm Snapdragon 821 MSM 8996 Pro in milliseconds (CPU metric).

TABLE I. COMPARISON OF CLASSIFIER ARCHITECTURES

Network architecture	Top-1 accuracy, %	Params, million	MAdds, million	CPU, ms
MobileNetV1	70.6	4.2	575	113
ShuffleNet(1.5)	71.5	3.4	292	-
ShuffleNet(x2)	73.4	5.4	524	-
NasNet-A	74.0	5.3	564	183
MobileNetV2	72.0	3.4	300	75
MobileNetV2(1.4)	74.7	6.9	585	143

Table 2 shows a comparative analysis of several detector architectures according to the following parameters [9, 10]:

- average accuracy on a COCO dataset (mAP metric);
- number of neural network parameters, millions (Params parameter);
- the number of operations of addition and multiplication, billions (parameter MAdds);
- the processing time for one color image from the COCO data set on a single core processor Qualcomm Snapdragon 821 MSM 8996 Pro in milliseconds (CPU metric).

TABLE II. COMPARISON OF DETECTOR ARCHITECTURES

Network architecture	mAP, %	Params, million	MAdds, billion	CPU, ms
SSD300	23.2	36.1	35.2	>1000
SSD512	26.8	36.1	99.5	>1000
YOLOv2	21.6	50.7	17.5	>1000
MobileNetV1+SSDLite	22.2	5.1	1.3	270
MobileNetV2+SSDLite	22.1	4.3	0.8	200

Based on an analysis of the accuracy and speed indicators of the most common architectures, it was decided to use the MobileNetV2 + SSDLite architecture to solve the problem of detection of license plate. The main advantages of this architecture:

- MobileNetV2 is used as a classifier which allows us to obtain high classification accuracy for a relatively small number of parameters;
- the combination of MobileNetV2 with the SSDLite detector is the fastest solution for detecting objects in the image at the moment.

The SSDLite architecture for the object detection task, using MobileNetV2 in the convolutional part, surpasses the well-known YOLOv2 real-time detector in accuracy on the MS COCO dataset, while showing 20 times faster and 10 times smaller size [11]. Figure 4 shows a comparison of the neural network architectures SSD and YOLO.

An image of any size is supplied to the input of the system, which is reduced to a size of 300 by 300 pixels and fed to the input of the neural network. The initial convolutional layers of the neural network extract features from the image, which are then processed by the detection procedure.

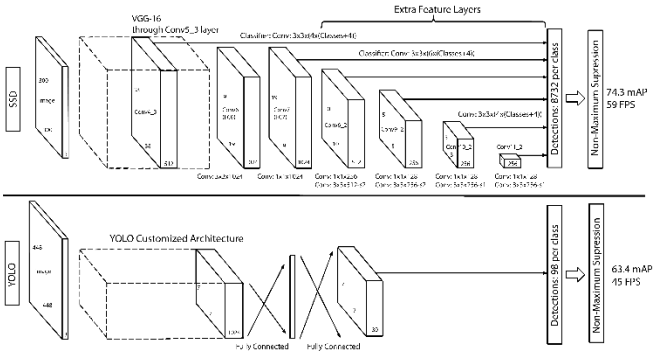

Fig. 4. Comparison of SSD and YOLO architectures.

Many convolutional networks predict the coordinates of bounding rectangles using fully connected layers based on feature extraction using convolutional layers. The constructed neural network, using only convolutional layers, predicts the offset and degree of confidence for the anchor box instead of predicting the absolute values of the coordinates and sizes of bounding rectangles [12-13].

Displacements are predicted at every place on the feature map since the prediction layer is convolutional. Using displacements instead of coordinates facilitates the training of a neural network. A similar principle is used in the Faster R-CNN local prediction module.

The use of reference areas slightly reduces the accuracy of the neural network, but significantly increases the completeness index, which allows to increase the result of the detection system as a whole [14]. The K-means method was used on the training data set to find the support areas. The number of reference regions equal to 4 was chosen as a compromise between the completeness index and the complexity of the model.

The output layer of the neural network combines features with both high and low resolution. A similar approach is used in the architecture of the ResNet neural network [15]. Thus, working with an expanded feature map allows us to detect both large and relatively small images of objects. The approach allows us to increase the quality of the detector. The output of a neural network is a feature map of size NxNx24. Each cell contains 6 predicted parameters for each of the 4 reference areas. The parameter p shows the confidence value, which is the ratio of the intersection to the union between the predicted area and the true value. Figure 5 shows the calculation of the intersection to union ratio.

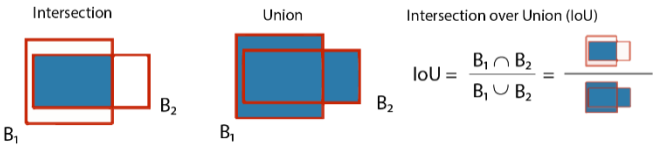

Fig. 5. Computing the intersection to union ratio.

Also, the probability of its belonging to one of the classes is predicted for each reference region. In our case it is either the “license plate” class (at the license plate detection stage) / “symbol” (at the text recognition stage) or the “background” class.

One of the problems encountered in detecting objects is the multiple detection of the same object or false positives of the detector. The No-Maximum Suppression algorithm is used to filter out unnecessary detections and leave only the area that most accurately describes the given object.

The algorithm for suppressing non-maxima discards objects whose probability of occurrence is lower than some given threshold value. The threshold value can be selected depending on the quality of the neural network, the training procedure, and various factors specific to the input images. This allows us to adapt the detection subsystem to various operating conditions. Further, the non-maximum suppression algorithm selects the most probable detection in a certain area and filters other detections of the same class that intersect strongly with the selected one.

The paper discusses the SSDLite architecture which uses MobileNetV2 as a feature extractor in the convolutional layers. Figure 6 demonstrates the architecture diagram of SSDLite with MobileNetV2.

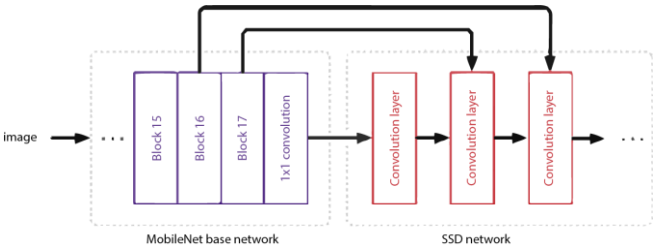

Fig. 6. Architecture diagram of SSDLite + MobileNetV2.

In the proposed architecture the results of several previous layers are submitted to the input of each Convolution layer of the SSDLite neural network in addition to the output value of the last layer of the MobileNet core network. The operation of MobileNet layers in this case is to convert the pixels of the input image into features that describe the content of the image for subsequent detection. SSD is designed independently of the core network, so it can work with almost everything, including MobileNet. Moreover, MobileNet + SSD uses an option called the SSDLite architecture, which uses depth-separated layers instead of the usual convolution to detect objects in the image. Using SSDLite on top of MobileNet2 makes it easy to get real-time recognition results even on a mobile device (from 30 FPS - the number of frames processed per second).

F. Training of the detector

In this work, the training of the license plate detector was carried out in 2 stages using the teaching methodology with transfer (transfer learning):

- pre-training on a general-purpose data set;
- retraining on a data set containing only images of the license plate recognition.

Both stages of training were conducted on a Google Compute Engine virtual machine with the following characteristics:

- 2 Intel Xeon Silver 41 processors;
- 64 GB RAM;
- 2 Nvidia Tesla K80 GPUs.

The first stage of training was carried out using the COCO dataset [16]. The main goal of this stage is to teach the neural network to distinguish low-level features of the image, such as lines, angles, texture properties, simple geometric shapes. For learning, The Tensorflow deep learning framework (a 48-hour learning process on the above virtual machine) was used for learning. The model had an average accuracy (mAP) of 20.9 achine based on the results of this training. Based on the results of this training, the model had an average accuracy (mAP) of 20.9.

The output layer of the pre-trained neural network contains 92 neurons (91 neurons correspond to output classes, and 1 neuron corresponds to the background class) since the COCO dataset contains objects belonging to 91 different classes. However, the detector should only localize objects belonging to the class "distributor". The output layer was removed from the neural network obtained after the first stage, and a new layer containing only 2 neurons was added instead of it (1 for the “license plate” class and 1 for the background) to do this. According to the transfer learning methodology, the remaining neural network weights are not fixed which will allow us to adjust them at the next stage for the correct detection of license plate.

The second stage of training was carried out using the following data sets:

- UFPR-ALPR, containing 4,500 color images of vehicles with Brazilian gas distribution devices measuring 1920 x 1080 pixels, obtained using GoPro Hero4 Silver, Huawei P9 Lite and iPhone 7 Plus cameras [17]. For each camera 1,500 images were taken, 900 of which were cars with gray license plates, 300 were cars with red license plates, 300 were motorcycles with gray license plates [17, 18].
- European License Plates, containing 10,000 photographs of the vehicles of EU countries of different sizes [14]. Types of license plates also vary.
- The database of the site avto-nomer.ru, containing 10,000 photos of Russian vehicles and 5,500 photos of vehicles in the CIS countries [19].
- Thus, the final data set consists of 30,000 photos of vehicles belonging to different countries. The set was randomly broken into 3 parts as follows:
- 6000 images (20% of the total) make up the test set used for final testing of the detector;
- 3000 images (10%) constitute a validation kit, which is used to check the accuracy of the detector in the learning process and make decisions about adjusting the parameters of the neural network;

- 21,000 images (70%) are used directly for training the neural network.

As the target metric was used Intersection over Union (IoU):

$$IoU = \frac{area(R\det \cap Rgt)}{area(R\det \cup Rgt)}$$

where Rdet is the predicted framing rectangle, Rgt is the true framing rectangle, area is the area calculation function [20].

The learning process was launched on the same virtual machine, the initial number of learning iterations (eras) was 1000, the criterion of early stopping the learning process was applied in the absence of growth of the target metric for 10 eras. Training was stopped by this criterion at the 47th epoch, the IoU value on the validation set at this moment is 0.93. Training of the license plate detector took 19 hours.

G. Character Detector Training

The MobileNetV2 + SSDLite architecture proved to be very good in the task of license plate detecting, so it was decided to use it in the task of detecting characters. The training was carried out in one step without the use of transfer learning since the detector should localize images with fairly simple attributes (symbols and numbers).

The training was carried out on the same dataset as the second stage of the training of the license plate detector (30,000 images of vehicles from different countries with different types of license plate). From each image of the vehicle, the framing rectangle of the license plates was cut out.

Since the character detector works with a large number of classes (26 characters of the Latin alphabet, 10 digits, background, a total of 37 classes) than the license plate detector, it was decided to split the initial data set in the following proportions:

- 6000 images (20%) make up the test suite;
- 6000 images (20%) constitute a validation set;
- 18,000 images (60%) make up the training set.

The average accuracy (mAP) is used as the target metric.

The learning process was launched on the same virtual machine, the initial number of epochs is 100, the criterion of early stopping of training was applied in the absence of growth of the target metric for 5 eras. Education was stopped by the criterion at the 18th epoch, the mAP value at this moment was equal to 0.54. The training time was 5 hours.

IV. Results and Discussion

A client-server application was developed, the screen form of which is shown in Figure 7.

It was used 6000 images which were postponed before the training stage of the detector for testing the recognition system.

This set contains the license plate recognition of the following countries:

- Russia, a total of 2000 images of various types;
- European Union, 2000 images of different countries;
- CIS, 1000 images of different countries;
- Brazil, 1000 images.

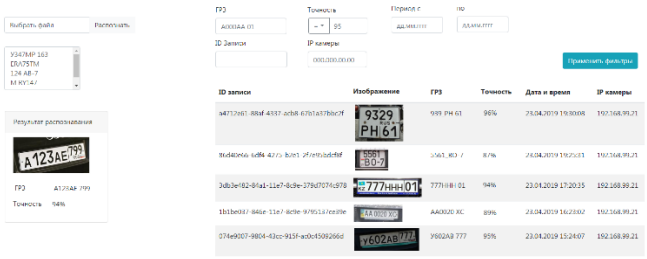

Fig. 7. Screen form of the developed application.

In addition, 1000 US numbers were added to the test suite to test the generalizability of neural networks to new types.

The following metrics were used to test the license plate detector:

- IoU;
- completeness (Recall) – a metric showing the proportion of the license plates found in the general data set;
- average detection time (DT), in milliseconds.

The following metrics were used to test the character detector:

- average detection accuracy (mAP);
- completeness (Recall2);
- average detection time (DT2) in milliseconds.

The following metrics were used to test the entire the system of license plate detector:

- the proportion of numbers in which more than 5 characters (5C) are correctly recognized;
- percentage of fully recognized numbers (AC).
- for testing, we used an instance of the Google Compute Engine virtual machine with the following characteristics:
    - 1 Intel Xeon Silver 4116 processor;
    - 16 GB RAM;
    - 100 GB of storage space on the SSD.

The test results are shown in table 3.

TABLE III. Comparison of Classifier Architectures

Dataset	Russia	EU	CIS	Brazil	USA
Number of images	2000	2000	1000	1000	1000
IoU	0.95	0.97	0.95	0.94	0.91
Recall	1	1	1	1	0.99
DT, ms	85	86	85	85	92
mAP, %	52	49	51	52	48
Recall2	0.99	0.98	0.99	0.99	0.91
DT2, ms	103	110	105	106	119
5C	0.99	0.99	0.99	0.99	0.97
AC	0.98	0.95	0.94	0.99	0.86

V. Conclusion

In the work the main stages of constructing a recognition system for license plate detector on digital images were considered. A comparative analysis of approaches to the construction of license plate recognition systems was carried out. The advantages and disadvantages of various detection and alignment algorithms, as well as approaches to constructing a feature space, are analyzed. During the analysis, the most appropriate neural network architecture was chosen, as well as the optimal learning method for different classes of tasks.

The subsystem for the license plate detection and recognition on digital images has been developed, and their interaction has been implemented within a single system. Data sets for training and testing the system were analyzed and selected as close as possible to the conditions of the real world.

Also, a web interface was implemented for working with recognition results and subsequent integration with other systems.

The developed system allows us to solve all the main classes of problems, such as detection and classification, according to the requirements of practice.

Successful testing of the license plate detection and recognition system, which showed resistance to changes in various factors that make it difficult to detect and recognize numbers, was carried out. The accuracy of the implemented system was at least 94%, which is an excellent result. Thus, the license plate detection and recognition system can be used to solve problems in practice

References

[1] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. Lawrence Zitnick and P. Dollár, “Microsoft COCO: Common Objects in Context,” arXiv preprint, arXiv: 1405.0312, 2015.

[2] Automatic License Plate Detection & Recognition using deep learning [Online]. URL: https://towardsdatascience.com/automatic-license-plate-detection-recognition-using-deep-learning-624def07eaaf.

[3] License Plate Recognition using OpenCV, YOLO and Keras [Online]. URL: https://medium.com/@theophilebuyssens/license-plate-recognition-using-opencv-yolo-and-keras-f5bfe03afc65.

[4] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L. C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks”, arXiv preprint, arXiv: 1801.04381, 2018.

[5] A. V. Nikonorov, M. V. Petrov, S. A. Bibikov, V. V. Kutikova, A. A. Morozov, N. L. Kazanskiy, “Image restoration in diffractive optical systems using deep learning and deconvolution”, Computer Optics, vol. 41, no. 6, pp. 875-887, 2017. DOI: 10.18287/2412-6179-2017-41-6-875-887.

[6] R. P. Bogush, I. Yu. Zakharova, “An algorithm for tracking people on video sequences using convolutional neural networks for indoor video surveillance,” Computer Optics, vol. 44, no. 1, pp. 109-116, 2020. DOI: 10.18287/2412-6179-CO-565.

[7] N. I. Chervyakov, P. A. Lyakhov, N. N. Nagornov, M. V. Valueva, G. V. Valuev, “Hardware implementation of a convolutional neural network using calculations in the system of residual classes”, Computer Optics, vol. 43, no. 5, pp. 857-868, 2019. DOI: 10.18287/2412-6179-2019-43-5-857-868.

[8] S. Du, M. Ibrahim, M. Shehata and W. Badawy, “Automatic license plate recognition (alpr): A state-of-the-art review,” IEEE Trans. Circuits Syst. Video Technol., vol. 23, no. 2, pp. 311-325, 2013.

[9] A Robust Real-Time Automatic License Plate Recognition Based on the YOLO Detector, arXiv preprint, arXiv: 1802.09567.pdf.

[10] License Plate Detection and Recognition in Unconstrained Scenarios [Online]. URL: http://openaccess.thecvf.com/content_ECCV_2018/papers/Sergio_Silva_License_Plate_Detection_ECCV_2018_paper.pdf.

[11] S. Yu, B. Li, Q. Zhang, C. Liu and M. Meng, "A novel license plate location method based on wavelet transform and emd analysis,” Pattern Recogn, vol. 48, no. 1, p. 114125, 2015.

[12] YOLOv3: An Incremental Improvement [Online]. URL: https://pjreddie.com/media/files/papers/YOLOv3. pdf.

[13] mAP (mean Average Precision) for Object Detection [Online]. URL: https://medium.com/@jonathan_hui/map-mean-average-precision-for-object-detection-45c121a31173.

[14] S. Chang, L. Chen, Y. Chung and S. Chen, “Automatic license plate recognition,” IEEE Trans. Intell. Transp. Syst., vol. 5, no. 1, p. 4253, 2004.

[15] B. Li, B. Tian, Y. Li and D. Wen, “Component-based license plate detection using conditional random field model,” IEEE Trans. Intell. Transp. Syst., vol. 14, no. 4, pp. 1690-1699, 2013.

[16] I. Giannoukos, C.-N. Anagnostopoulos, V. Loumos and E. Kayafas, “Operator context scanning to support high segmentation rates for real time license plate recognition,” Pattern Recogn., vol. 43, no. 11, p. 38663878, 2010.

[17] W. Zhou, H. Li, Y. Lu and Q. Tian, “Principal visual word discovery for automatic license plate detection,” IEEE Trans. Image Process., vol. 21, no. 9, pp. 4269-4279, 2012.

[18] C. Anagnostopoulos, I. Anagnostopoulos, V. Loumos and E. Kayafas, “A license plate-recognition algorithm for intelligent transportation system applications,” IEEE Trans. Intell. Transp. Syst., vol. 7, no. 3, pp. 377-392, 2006.

[19] Y. Yuan, W. Zou, Y. Zhao, X. Wang, X. Hu and N. Komodakis, “A robust and efficient approach to license plate detection,” IEEE Trans. Image Process., vol. 26, no. 3, pp. 1102-1114, 2017.

[20] G. Hsu, J. Chen and Y. Chung, “Application-oriented license plate recognition,” IEEE Trans. Veh. Technol., vol. 62, no. 2, pp. 552-561, 2013.