jet INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2023, VOL. 69, NO. 2, PP. 275-280
Manuscript received December 31, 2022; revised April, 2023. DOI: 10.24425/ijet.2023.144361

An Approach to License Plate Recognition in Real Time Using Multi-stage Computational Intelligence Classifier

Michał Kekez

***Abstract*—Automatic car license plate recognition (LPR) is widely used nowadays. It involves plate localization in the image, character segmentation and optical character recognition. In this paper, a set of descriptors of image segments (characters) was proposed as well as a technique of multi-stage classification of letters and digits using cascade of neural network and several parallel Random Forest or classification tree or rule list classifiers. The proposed solution was applied to automated recognition of number plates which are composed of capital Latin letters and Arabic numerals. The paper presents an analysis of the accuracy of the obtained classifiers. The time needed to build the classifier and the time needed to classify characters using it are also presented.**

***Keywords*—car license plates; LPR; ANPR; OCR; image processing; neural network; Random Forest**

I. Introduction

LICENSE plate recognition (LPR) systems, also known as automatic number plate recognition (ANPR), use image processing and optical character recognition to obtain vehicle's license plate number from camera images. Due to digital camera market boom in the late 1990s and early 2000s [1] the above-mentioned systems are used increasingly in recent years. Their applications include, but are not limited to: electronic toll collection on roads and parking lots, verification of vignette payment on motorway, verification of payment in city's paid parking zone, access control at parking lot barriers, measurement of time spent in parking system, and law-enforcement applications (e.g. calculation of the vehicle average speed on a given section of the road).

The intelligent transportation systems (ITS) can use images from traffic cameras for [1]: license plate recognition (LPR), car make and model recognition (MMR), and car color recognition (CR). The obtained data can be used in ITS to monitor traffic intensity and to determine vehicle routes, i.e. to determine zones within which the source and destination of the journey are located. Gathered information can be used for dynamic modelling and applied to dynamic traffic signal control, dynamic tolls, and traffic flow forecasting [2].

Various computational intelligence methods are used for LPR. Convolutional neural networks (CNN) are increasingly used for image recognition, including LPR and optical character recognition (OCR). In [3], a deep learning-based tool, YOLOv4 [4], was used to vehicle type (VT) and license plate recognition. YOLOv4 was also used for fast, low-latency pedestrian detection in autonomous driving [5], while previous versions (YOLOv2, YOLOv3) were applied for LPR many times, e.g. in [6]. Another deep learning-based tool, SSD, was used for detecting objects in images [7]. The drawbacks of deep-learning tools are: large computing power requirements, moderate speed in some applications [3], and possibility of adversarial attack [8]. However, the accuracy of recognition of license plate characters based on deep learning ranged from 85.45% to 99.92% [9].

Various LPR techniques were discussed in [9],[10]. Cascade structure with AdaBoost learning was used for LPR in [11]. Application of k-nearest neighbors method for LPR was described in [12].

All above-mentioned LPR algorithms require a sufficiently large training data set consisting of license plate images. Problem of small number of training examples can be solved by using Generative Adversarial Networks (GAN) method to generate license plate images [13].

II. License Plate Recognition Process

License plate recognition consists of several steps, presented in Fig. 1. Image from camera is processed in two main stages: license plate detection (i.e. finding its location within the image) and license plate recognition.

A. License plate detection

The process of detecting the license plate (LP) in the image depends on the location of the camera. If the camera is located near barrier at the entrance to the car par, then only one vehicle is visible and only one LP must be found. Cameras placed over multi-lane thoroughfares can generate large images covering all lanes and more than one LP in each lane [3]. Their field of view may also be limited to a smaller fragment of one lane, with one LP visible [1].

According to [9], classical computer vision techniques include (Fig. 1): edge-based methods (vertical or horizontal edge detection [14], usually using Sobel filter [15]), color-based methods (searching for unique color combination of LP and its letters and utilizing histogram intersection [16], mean shift algorithm [17], genetic algorithms, or fuzzy systems [18]), texture-based methods (looking for unique pixel intensity distribution around the LP region and utilizing scan-line techniques [19], vector quantization [20], sliding concentric

Michał Kekez is with Kielce University of Technology (e-mail: mkekez@tu.kielce.pl).

window [21], Gabor filter [22], or wavelet transform [23]), and character-based methods (including scale-space analysis and region-based approach with use of neural networks [24]).

Statistical classifiers that are used for LP detection utilize [9]: AdaBoost [25], Support Vector Machine (SVM), or SVM in connection with Continuously Adaptive Mean Shift (CAMShift).

Deep learning was also used for LP localization, e.g. in [3],[26].

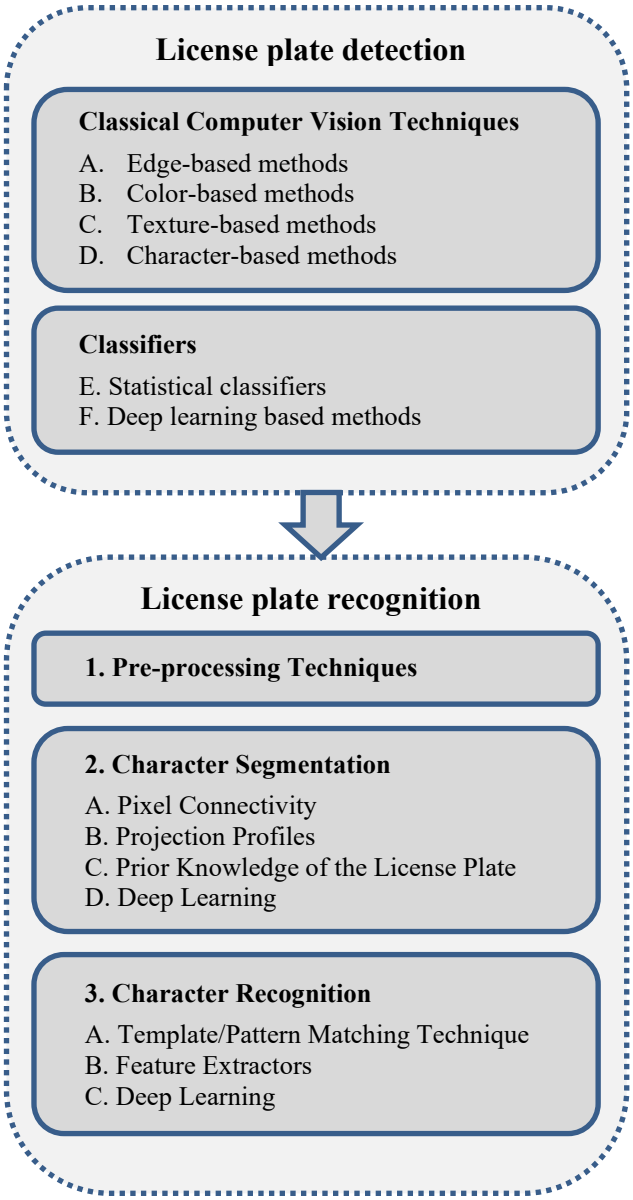


Fig. 1. License plate recognition process ([9], modified)

B. *License plate recognition*

Recognition of the LP consists in analyzing a section of the image from the camera, containing the license plate. First, preprocessing is performed, and then segmentation is performed to extract the areas occupied by individual characters. Each of the areas is subjected to optical character recognition (OCR).

1) Pre-processing techniques

Depending on the angle between the camera and the license plate, affine transformations of the image – scale, rotate, shear – may be necessary to avoid geometric distortion of the LP image. In the image thus obtained, the edges of the license plate are parallel to the corresponding edges of the image. Appropriate license plate image is also not "stretched" or "squeezed".

In the system described in [1], during the pre-processing step, the image is also converted to a grayscale (Fig. 2b), then blurred with Gaussian filter, and finally filtered by application of noise removal morphological operations [1].

Later, binarization is performed using Otsu algorithm [27] or using Otsu method combined with dilation morphological operation [1]. In some LPR systems, the image is then negated (Fig. 2c).

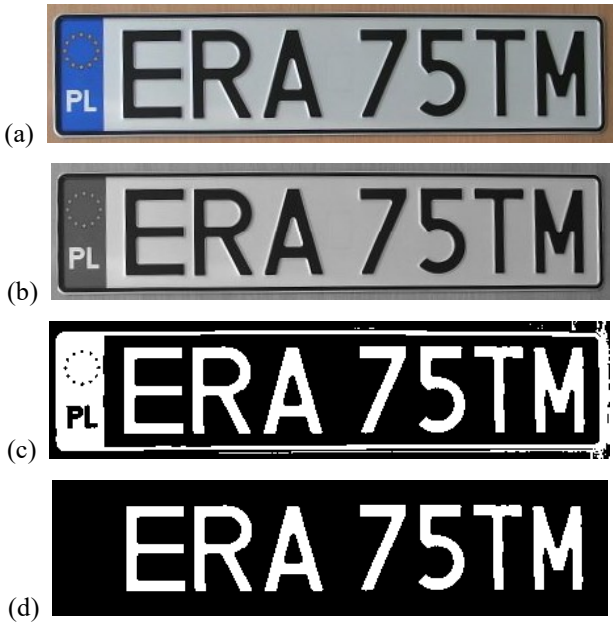


Fig. 2. Preprocessing of the LP image: (a) original image [28], (b) grayscale image, (c) binary image (Otsu algorithm), negated, (d) image after removing the LP frame

The frame surrounding the white license plate is removed from the image in various ways (Fig. 2d). One of them is application of Canny Edge Detector followed by the selected contour extraction method [1]. After this step, image contains only dark LP digits and letters on the light background [1] (or inversely, if the image was previously negated).

2) Character segmentation

Segmentation involves extracting separate images of each number and letter from the LP image. In general, character segmentation stage may require one of the following methods [9]: pixel connectivity (finding connected components) [29], projection profiles [30], prior knowledge of the LP properties [31], and deep learning [3],[32],[33].

In the simplest version, each connected component is treated as a separate character (Fig. 3). In the enhanced version [1], an adaptive binarization procedure is performed, which determines binarization threshold depending on the neighborhood of successive pixel [1].

In some cases, shown in Fig. 4, (e.g. unfavorable lighting conditions, a small leaf on LP, scratched part of the letter, or small obstacles between the camera and LP) the image must be enhanced by morphological operations such as dilation or erosion.

Fig. 3. Character segmentation. Individual characters are marked with a green frame.

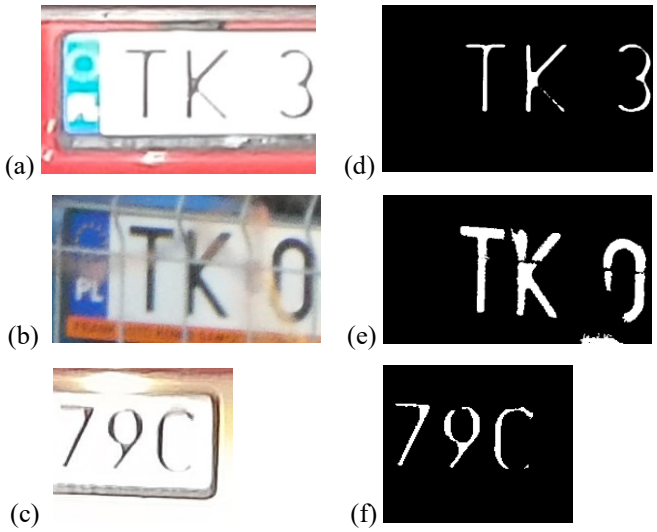


Fig. 4. Fragments of images that require dilation before character segmentation: (a-c) original images, (d-f) images just before segmentation

As a result of segmentation, each letter or digit is obtained as a separate image (object). Objects too wide or too high, not containing LP characters, must be filtered out [1].

3) Character recognition

Each separate object containing one character is recognized, using a method belonging to one of the these groups [9]: pattern matching, feature extractors, deep learning. The process of character recognition is often called Optical Character Recognition (OCR).

Pattern matching is the simplest method, using similarity measures between the pattern and the tested character [34].

Feature extractors (eigenvector transformation [35], Gabor filter [36], Kirsh edge detection [37], or others) produce a set of values which is then fed into the classifier input. The classifier is built using machine learning or computational intelligence methods, e.g. Support Vector Machines (SVM) [38] or Hidden Markov Model (HMM) [39].

Deep learning based systems use convolutional neural networks (CNN) [32], mainly using YOLO detector [33].

In [1], Tesseract OCR software [40] was used. Tesseract OCR libraries are also used in OpenALPR software [41], dedicated to license plate recognition.

III. Proposed Methodology

The aim of the article was to develop an approach for license plate recognition (LPR) that allows for a quick construction of a classifier based on a small training set (containing only one record for each character). Another assumption was the possibility of building a training set for a given country's license plates without using any actual images, and by using only publicly available sources like legal acts which describe shape and dimensions of letters and digits used on LP (only the validation of the classifier is carried out on large sets of camera images).

Thanks to this, it is possible to build a system consisting of parallel operating classifiers (one classifier for license plates from one country) that will recognize license plates from many countries, even without using real photographs of license plates from these countries.

The main part of the proposed approach is optical character recognition (OCR), however it requires images of characters obtained in the previous step.

A. Detection of license plate, pre-processing and character segmentation

The proposed system assumes that the camera's field of view is limited to a fragment of one lane with one LP visible or the camera is located near barrier at the entrance to the car park, where only one LP is visible.

The proposed approach can use any license plate detection method. The image containing the detected license plate may also cover the area around the plate, provided that the height of the letters and numbers is at least 30% of the height of the image. If the license plate was not perpendicular to the optical axis of the camera, affine image transformations (scale, rotate, shear) are carried out.

In the pre-processing stage, the image is converted to grayscale, binarized, and negated. License plate frame and objects outside the license plate are removed. Segmentation is carried out by finding connected components, whereby optional dilation or erosion may be performed.

B. Construction of the classifier for character recognition

In the proposed approach, a classifier, which is a cascade of neural network and several parallel Random Forest/classification tree/rule list classifiers, is used for character recognition. The methodology for its creation is described below.

1) Training data

The training data for the classifier for Polish license plates contains records consisting of values of one output attribute y (*character*) and 11 input attributes: x_1 ($p01$), x_2 ($v1$), x_3 ($v2$), x_4 ($v3$), x_5 ($h1$), x_6 ($h2$), x_7 ($h3$) , x_8 ($o1$) , x_9 ($o2$) , x_{10} ($a1$) , x_{11} ($a2$).

For the image of each digit and capital letter (25 characters without the letter Q) appearing in Polish license plates, the values of eleven descriptors, named $p01$, $v1$, $v2$, $v3$, $h1$, $h2$, $h3$, $o1$, $o2$, $a1$, $a2$ were calculated.

The $p01$ descriptor is the ratio of the width to the height of the smallest rectangle containing a letter or number. The calculated value is rounded to 1 decimal place, and if the rounding error exceeds 0.0333, two versions are generated, one

with rounding down and one with up. Moreover, if one letter can appear in two versions with slightly different proportions (e.g. on Polish registration plates before and after June 30, 2018 [42]), the $p01$ value is calculated separately for each version. Hence there can be 2 or even 4 records for one letter or number in the training set.

The $v1$, $v2$, or $v3$ descriptor is the maximum number of "on" (white) pixels in one vertical line (column) of the image, divided by the image height, but calculated for columns lying in the appropriate ranges: between 0 and w for $v1$, between (*width*–w)/2 and (*width*+w)/2 for $v2$, and between *width*–w and *width* for $v3$, where *width* is the image width, and w is 1/6 of the letters' and digits' height on license plates (common for all characters). The column ranges for $v1$, $v2$, and $v3$ are shown in Fig. 5a as the green frames.

The $h1$, $h2$, or $h3$ descriptor is the maximum number of "on" pixels in one horizontal line (row) of the image, divided by the image width, but calculated for rows lying in the ranges: between 0 and w for $h1$, between (*height*–w)/2 and (*height*+w)/2 for $v2$, and between *height*–w and *height* for $v3$, where *height* is the image height. The column ranges for $h1$, $h2$, and $h3$ are shown in Fig. 5b as the green frames.

The $o1$ and $o2$ descriptors are calculated in a similar way to $v1$ and $v3$, respectively, but lines are oblique (Fig. 6c and Fig. 6d).

The $a1$ and $a2$ descriptors (Fig. 6e and Fig. 6f) are equal to the ratio of number of "on" pixels in a given area to the number of all pixels in that area. The area for $a1$ consists of pixels in rows between 0 and w, and, in the same time, in columns between 0 and w. The area for $a2$ covers pixels in the same rows as the area for $a1$ descriptor, but columns between *width*-w and *width*.

The proposed descriptors are slightly similar to horizontal or vertical projections or zoning density used in [43].

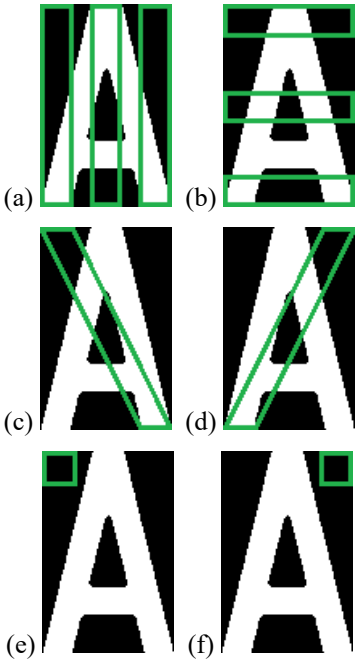


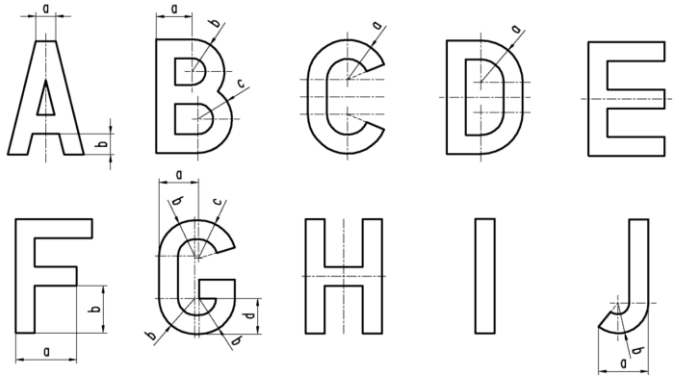
Fig. 5. Location of the image regions used for the calculation of: (a) $v1$, $v2$, $v3$, (b) $h1$, $h2$, $h3$, (c) $o1$, (d) $o2$, (e) $a1$, and (f) $a2$ descriptors.

For the construction of the above mentioned training data, shapes and sizes of letters and digits were used according to [42], as shown in Fig. 6.

2) Building the classifier

Two-stage classifiers that use over 100 descriptors of the image containing a letter or a number are used in license plate recognition, e.g. for Chinese license plates [43]. The proposed classifier consist of three stages (layers) and uses only 11 descriptors.

In the first stage, a multilayer perceptron with one hidden layer is built using the training dataset D_1 described above, but without the last 4 input attributes (x_8, x_9, x_{10}, x_{11}).



A	a	9,25
	b	15
B	a	23
	b	R21
	c	R23,5
C	a	R25
D	a	R26
F	a	38
	b	35,5
G	a	24,5
	b	R22,5
	c	R23
	d	28
J	a	30
	b	R19

Fig. 6. A fragment of the definition of sizes and shapes of letters and digits used in Polish car license plates [42].

Then, dataset D_1 is modified: random noise is added to the character images and slightly skewed character images are created (alternatively, real license plates images can be used instead). This dataset, D_2, is then classified using the classifier built in the first stage. Disjoint sets of characters, S_1, S_2, …, S_n, that were indistinguishable for this classifier (i.e. at least once misclassified as another character from the same set) are created. In the second stage, a separate training dataset $D_{2,i}$ (i=1,2,…,n) is created for each such set of characters, S_i. The $D_{2,i}$ consist of all records from D_2 that describe the characters belonging to the set S_i. Then, for each set S_i, the random forest [44] classifier, classification tree, or list of rules is built using $D_{2,i}$ dataset containing values of all 11 input attributes (x_1, x_2, …, x_{11}).

In the third step, the rules for restrictions on entire registration plate are applied. For example: if character is classified as '0' or 'O', and its position on license plate is 4, 5, 6, or 7, and the second character in the license plate is a letter, then the character is 0.

IV. Results and Discussion

The proposed methodology was used to build a system for recognizing Polish car license plates. First, a training set D_1, containing 58 records describing 35 letters and numbers, was built. Using the MultilayerPerceptron function in Weka software [45], an artificial neural network with 21 nodes in the hidden layer and 35 nodes in the output layer was created in 6.02 seconds. This stage 1 classifier correctly classified 100% cases from the training set (Table I).

Then, a training set D_2, containing 226 records describing characters from 32 artificially generated or real license plates, was classified by stage 1 classifier with accuracy of 88.1%. Based on the classification results, the following characters sets were determined: S_1 = {'A', 'G', 'K', 'V', 'X', '6'} , S_2 = {'B', '8'}, S_3 = {'H', 'M', 'N', '1'}, S_4 = {'O', '0'}, S_5 = {'P', 'R', '9'}, S_6 = {'S', '3', '5'}.

Table I
Classification accuracy

	Training set D_1	Validation set
After stage 1	100%	88.7%
After stage 2	-	98.7%
After stage 3	-	99,0%

Next, stage 2 classifiers were built (some of them in the form of lists of rules, shown in Fig. 7).

```
if stage1_output in ['P','R','9']:
    if v1 > 0.88 and o1 <= 0.61: char='P'
    if v1 > 0.88 and o1 > 0.61: char='R'
    if v1 <= 0.88: char='9'
if stage1_output in ['S','3','5']:
    if a1<=0.70 and v1<=0.49: char = '3'
    if a1<=0.70 and v1>0.49: char = 'S'
    if a1>0.70: char = '5'
```

Fig. 7. Rules for S_5 and S_6 sets, built in stage 2.

The stage 3 classifier consists of the rules shown in Fig. 8.

```
if stage2_output in ['B','D','I','O','Z'] and 4<=position<=8:
    if letter_at_position_2 and stage2_output=='B': char = '8'
    if letter_at_position_2 and stage2_output=='D': char = '0'
    if letter_at_position_2 and stage2_output=='I': char = '1'
    if letter_at_position_2 and stage2_output=='O': char = '0'
    if letter_at_position_2 and stage2_output=='Z': char = '2'
else:
    char = stage2_output
```

Fig. 8. Rules forming the stage 3 classifier.

A new dataset consisting of 390 letters and digits from 52 license plates was classified using stage 1, 2, and 3 classifiers. After stage 1, 88.7% of characters were correctly classified. After stage 2, the percentage of correctly classified characters increased to 98.7%, and after stage 3 one 'O' was corrected to '0', slightly improving the overall result (Table I).

The calculation time for a single license plate image, already cut out from the image from camera, carried out on a Pentium N4200@1.1GHz computer, is presented in Table II.

Table II
Calculation time for a single license plate

	Time for a typical license plate (7 characters), in milliseconds
LP image processing in Matlab	85 ms
Descriptors calculation in Matlab	49 ms
Classification, stage 1	< 1 ms
Classification, stage 2	< 1 ms
Classification, stage 3	< 1 ms

Conclusion

The advantage of the proposed approach is the construction of the classifier using a small set of data (one record in the training set for one character). The classifier created in stage 1 without training data from the cameras, and only based on information about the shape and size of letters used in a given country, achieved an accuracy of 100% on the training set and about 88% on the test set consisting of 52 license plates. Accuracy was increased to over 98% by creating additional classifiers in step 2. However, further validation on larger datasets and likely changes to stage 2 classifiers are needed.

The proposed approach allows for the quick construction of a system consisting of parallel classifiers that will recognize license plates from many countries applying the Vienna Convention on Road Traffic. An additional advantage is the time of building individual classifiers, not exceeding a few seconds, which is a great advantage compared to popular methods using deep learning. The processing time of the entire system depends mainly on the image processing time, because the calculations performed by the classifier take less than 1 millisecond.

References

[1] R. Baran, T. Rusc, P. Fornalski, "A smart camera for the surveillance of vehicles in intelligent transportation systems", Multimedia Tools and Applications, vol. 7, pp. 10471–10493, 2016: https://doi.org/10.1007/s11042-015-3151-y

[2] W. Y. Szeto "Dynamic modeling for intelligent transportation system applications", Journal of Intelligent Transportation Systems, vol. 18, no. 4, pp. 323–326, 2014. https://doi.org/10.1080/15472450.2013.834770

[3] S.-H. Park, S.-B. Yu, J-A. Kim, H. Yoon, "An All-in-One Vehicle Type and License Plate Recognition System Using YOLOv4", Sensors, vol. 22, no. 3, 921, 2022. https://doi.org/10.3390/s22030921

[4] A. Bochkovskiy, C. Y. Wang, H. Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection", arXiv 2020, arXiv:2004.10934.

[5] K. Roszyk, M. R. Nowicki, P. Skrzypczyński, "Adopting the YOLOv4 Architecture for Low-Latency Multispectral Pedestrian Detection in Autonomous Driving", Sensors, vol. 22, no. 3, 1082, 2022. https://doi.org/10.3390/s22031082

[6] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection", 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779-788, 2016. https://doi.org/10.1109/CVPR.2016.91

[7] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, "SSD: Single shot MultiBox detector", in Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016, Springer International Publishing: Cham, Switzerland, pp. 21–37, 2016. https://doi.org/10.1007/978-3-319-46448-0_2

[8] Ł. Grad, Adversarial Uncertainty Learning in Deep Neural Networks, University of Warsaw, seminar, October 29, 2021,

https://www.mimuw.edu.pl/sites/default/files/seminaria/adversarial_uncertainty_in_deep_learning-2.pdf

[9] J. Shashirangana, H. Padmasiri, D. Meedeniya, C. Perera, "Automated License Plate Recognition: A Survey on Methods and Techniques", IEEE Access, vol. 9, pp. 11203–11225, 2020. https://doi.org/10.1109/ACCESS.2020.3047929

[10] C. E. Anagnostopoulos, I. E. Anagnostopoulos, I. D. Psoroulas, V. Loumos, E. Kayafas, "License Plate Recognition From Still Images and Video Sequences: A Survey", IEEE Trans. Intell. Transp. Syst., vol. 9, pp. 377–391, 2008. https://doi.org/10.1109/TITS.2008.922938

[11] B.-G. Han, J. T. Lee, K.-T. Lim; Y. Chung, "Real-time license plate detection in high-resolution videos using fastest available cascade classifier and core patterns", ETRI J., vol. 37, pp. 251–261, 2015. https://doi.org/10.4218/etrij.15.2314.0077

[12] S. Park, H. Yoon, S. Park, "Multi-style license plate recognition system using k-nearest neighbors", KSII Trans. Internet Inf. Syst. (TIIS), vol. 13, pp. 2509–2528, 2019. https://doi.org/10.3837/tiis.2019.05.015

[13] B.-G. Han, J.T. Lee, K.-T. Lim, D.-H. Choi, "License plate image generation using generative adversarial networks for end-to-end license plate character recognition from a small set of real images", Applied Sciences, vol. 10, 2780, 2020. https://doi.org/10.3390/app10082780

[14] G. Heo, M. Kim, I. Jung, D.-R. Lee, I.-S. Oh, "Extraction of car license plate regions using line grouping and edge density methods", in Proc. Int. Symp. Inf. Technol. Converg. (ISITC), 23-24 Nov. 2007, pp. 37–42, 2007. https://doi.org/10.1109/ISITC.2007.79

[15] R. C. Gonzalez, R. E. Woods, "Digital Image Processing", third edition, 3rd Edition, Pearson Prentice Hall, Upper Saddle River, 2008.

[16] W. Jia, H. Zhang, X. He, and Q. Wu, "Gaussian weighted histogram intersection for license plate classification", in Proc. 18th Int. Conf. Pattern Recognit. (ICPR), vol. 3, pp. 574–577, 2006. https://doi.org/10.1109/ICPR.2006.596

[17] W. Jia, H. Zhang, X. He, and M. Piccardi, "Mean shift for accurate license plate localization", in Proc. IEEE Intell. Transp. Syst., 13-16 Sep. 2005, pp. 566–571, 2005. https://doi.org/10.1109/ITSC.2005.1520110

[18] F. Wang, L. Man, B. Wang, Y. Xiao, W. Pan, and X. Lu, "Fuzzy-based algorithm for color recognition of license plates", Pattern Recognition Letters, vol. 29, no. 7, pp. 1007–1020, 2008. https://doi.org/10.1016/j.patrec.2008.01.026

[19] H.-K. Xu, F.-H. Yu, J.-H. Jiao, and H.-S. Song, "A new approach of the vehicle license plate location", in Proc. 6th International Conference on Parallel and Distributed Computing Applications and Technologies (PDCAT), pp. 1055–1057, 2005. https://doi.org/10.1109/PDCAT.2005.24

[20] R. Zunino, S. Rovetta, "Vector quantization for license-plate location and image coding", *IEEE Trans. Ind. Electron.*, vol. 47, no. 1, pp. 159–167, Feb. 2000. https://doi.org/10.1109/41.824138

[21] 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, no. 3, pp. 377–392, Sep. 2006. https://doi.org/10.1109/TITS.2006.880641

[22] H. Caner, H. S. Gecim, and A. Z. Alkar, "Efficient embedded neuralnetwork-based license plate recognition system", IEEE Transactions on Vehicular Technology, vol. 57, no. 5, pp. 2675–2683, Sep. 2008. https://doi.org/10.1109/TVT.2008.915524

[23] C.-T. Hsieh, Y.-S. Juan, and K.-M. Hung, ''Multiple license plate detection for complex background,'' in Proc. 19th Int. Conf. Adv. Inf. Netw.Appl. (AINA), vols. 1–2, pp. 389–392, 2005. https://doi.org/10.1109/AINA.2005.257

[24] J. Matas and K. Zimmermann, "Unconstrained licence plate and text localization and recognition", in Proc. IEEE Intell. Transp. Syst., Sep. 2005, pp. 225–230, 2005. https://doi.org/10.1109/ITSC.2005.1520111

[25] X. Zhang, P. Shen, Y. Xiao, B. Li, Y. Hu, D. Qi, X. Xiao, and L. Zhang, "License plate-location using AdaBoost algorithm", in Proc. IEEE International Conference on Information and Automation, Jun. 2010, pp. 2456–2461, 2010. https://doi.org/10.1109/ICINFA.2010.5512276

[26] Z. Selmi, M. B. Halima, A. M. Alimi, ''Deep learning system for automatic license plate detection and recognition,'' in Proc. 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), 9-15 November 2017, pp. 1132–1138, 2017. https://doi.org/10.1109/ICDAR.2017.187

[27] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms", IEEE Transactions on Systems, Man, and Cybernetics, vol. 9, no. 1, pp. 62–66, 1979. https://doi.org/10.1109/TSMC.1979.4310076

[28] Polish license plate from Radomsko poviat https://upload.wikimedia.org/wikipedia/commons/9/90/Pltableseries2006.jpg. Viewed 29 December 2022.

[29] T. Nukano, M. Fukumi, M. Khalid, "Vehicle license plate character recognition by neural networks", in Proc. Int. Symp. Intell. Signal Process. Commun. Syst. (ISPACS), pp. 771–775, 2004. https://doi.org/10.1109/ISPACS.2004.1439164

[30] S. Du, M. Ibrahim, M. Shehata, W. Badawy, ''Automatic license plate recognition (ALPR): A state-of-the-art review,'' IEEE Transactions on Circuits and Systems for Video Technology, vol. 23, no. 2, pp. 311–325, 2013. https://doi.org/10.1109/TCSVT.2012.2203741

[31] C. Busch, R. Domer, C. Freytag, H. Ziegler, "Feature based recognition of traffic video streams for online route tracing", in Proc. 48th IEEE Vehicular Technology Conference. Pathway Global Wireless Revolution (VTC), 21 May 1998, vol. 3, pp. 1790–1794, 1998. https://doi.org/10.1109/VETEC.1998.686064

[32] S. Montazzolli, C. Jung, "Real-time Brazilian license plate detection and recognition using deep convolutional neural networks", in Proc. 30th SIBGRAPI Conf. Graph., Patterns Images, Oct. 2017, pp. 55–62, 2017. https://doi.org/10.1109/SIBGRAPI.2017.14

[33] R. Laroca, L. A. Zanlorensi, G. R. Gonçalves, E. Todt, W. R. Schwartz, D. Menotti, "An efficient and layout-independent automatic license plate recognition system based on the YOLO detector", *IET Intell Transp Syst*. vol. 15, pp. 483– 503, 2021. https://doi.org/10.1049/itr2.12030

[34] C. A. Rahman, W. Badawy, A. Radmanesh, "A real time vehicle's license plate recognition system", in Proc. IEEE Conference on Advanced Video and Signal Based Surveillance, Miami 22 July 2003, pp. 163–166, 2003. https://doi.org/10.1109/AVSS.2003.1217917

[35] H. A. Hegt, R. J. de la Haye, N. A. Khan, "A high performance license plate recognition system", in Proc. IEEE International Conference on Systems, Man, and Cybernetics (SMC), 14 Oct. 1998, pp. 4357–4362, 1998. https://doi.org/10.1109/ICSMC.1998.727533

[36] P. Hu, Y. Zhao, Z. Yang, J. Wang, "Recognition of gray character using Gabor filters", in Proc. 5th Int. Conf. Inf. Fusion (FUSION), vol. 1, pp. 419–424, 2002. https://doi.org/10.1109/ICIF.2002.1021184

[37] S. N. H. S. Abdullah, M. Khalid, R. Yusof, K. Omar, "License plate recognition using multi-cluster and multilayer neural networks", in Proc. 2nd Int. Conf. Inf. Commun. Technol., vol. 1, pp. 1818–1823, 2006. https://doi.org/10.1109/ICTTA.2006.1684663

[38] K. K. Kim, K. I. Kim, J. B. Kim, and H. J. Kim, "Learning-based approach for license plate recognition", in Proc. Neural Netw. Signal Process. X, IEEE Signal Process. Soc. Workshop, vol. 2, pp. 614–623, 2000. https://doi.org/10.1109/NNSP.2000.890140

[39] D. Llorens, A. Marzal, V. Palazón, J. M. Vilar, "Car license plates extraction and recognition based on connected components analysis and HMM decoding", In: Marques, J.S., Pérez de la Blanca, N., Pina, P. (eds) Pattern Recognition and Image Analysis. IbPRIA 2005. Lecture Notes in Computer Science, vol. 3522, pp. 571–578, Springer, Berlin, Heidelberg, 2005. https://doi.org/10.1007/11492429_69

[40] Tesseract. https://github.com/tesseract-ocr. Viewed 29 December 2022.

[41] OpenALPR. https://github.com/openalpr/openalpr. Viewed 29 December 2022.

[42] Rozporządzenie Ministra Infrastruktury i Budownictwa z dnia 11 grudnia 2017 r. w sprawie rejestracji i oznaczania pojazdów oraz wymagań dla tablic rejestracyjnych (Polish: Regulation of the Minister of Infrastructure and Construction of 11 December 2017 on the registration and marking of vehicles and requirements for license plates)

[43] X. Pan, X. Ye, S. Zhang, "A hybrid method for robust car plate character recognition", Engineering Applications of Artificial Intelligence, vol. 18, no. 8, pp. 963–972, 2005. https://doi.org/10.1016/j.engappai.2005.03.011

[44] L. Breiman, "Random Forests", Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[45] E. Frank, M. A. Hall, I. H. Witten, "The WEKA Workbench". Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques", Morgan Kaufmann, Fourth Edition, 2016.