



UDC 004.032.26, 528.854

LICENSE PLATE RECOGNITION SYSTEM BASED ON MASK R-CNN

Onishchenko D.¹, Liubchenko N.², Podorozhniak A.³^{1,2,3}National Technical University – «Kharkiv Polytechnic Institute»ORCID: ¹<https://orcid.org/0000-0002-4783-2053>, ²<https://orcid.org/0000-0002-4575-4741>,³<https://orcid.org/0000-0002-6688-8407>E-mail: ¹onishchnko21@gmail.com, ²nataliia.liubchenko@kphi.edu.ua, ³andrii.podorozhniak@kphi.edu.ua

Copyright © 2023 by author and the journal “Automation of technological and business – processes”.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0>

DOI: 10.15673/atbp.v15i3.2623

Abstract. Automatic license plate recognition (ALPR) systems can be found in many different applications. If a person has a driving license, the person probably had already seen smart road camera after a speed limit sign or on a crossroad. Every year number of cars on roads growth very fast. It is also obvious that such systems can be used out-of-road situations. For instance, this type of systems can be used for automatic access control on private property or smart parking, or even log system that are being used literally everywhere. Because of popularity of ALPR systems, there are two main goals, which are being pursued by researches: speed and accuracy of recognition. Speed of the detection is important for real-time systems. Accuracy is important for every system. The more accurate a system is, the more reliable it is. For example, car accident detection systems should be as accurate as possible in order to be used, because no one wants to get billed with the wrongdoing, that wasn't committed by them.

The purpose of the study is to develop high precision automatic license plate detection system with number extraction possibilities. In order to achieve the goal many different modern solutions and technologies were studied and solution is presented. The main technology of the project is artificial intelligence system and, more specifically, convolutional neural network. As the main algorithm Mask R-CNN is used for license plate detection. To present reasonable research, the system was tested on different hardware (CPU, GPU, Raspberry PI 4) and on different datasets.

Анотація. Системи автоматичного розпізнавання номерних знаків (АРНЗ) можна знайти в різних сферах людської діяльності. Якщо у людини є водійське посвідчення, то, напевно, вона вже бачила розумну дорожню камеру за знаком обмеження швидкості або на перехресті. З кожним роком кількість автомобілів на дорогах дуже стрімко зростає. Також очевидно, що такі системи можна використовувати поза межами доріг. Наприклад, цей тип систем можна використовувати для автоматичного контролю доступу на приватну власність або розумну парковку, або навіть для систем ведення обліку, які використовуються буквально всюди. Через популярність систем АРНЗ існує дві основні цілі, які переслідують дослідження: швидкість і точність розпізнавання. Швидкість виявлення важлива для систем реального часу. Точність важлива для кожної системи. Чим точніша система, тим вона надійніша. Наприклад, системи виявлення автомобільних аварій мають бути максимально точними, щоб їх можна було використовувати, оскільки ніхто не хоче отримати рахунок за порушення, скоєні не ним.

Метою дослідження є розробка високоточної автоматичної системи виявлення номерних знаків з можливістю вилучення номера. Для досягнення мети було вивчено багато різних сучасних рішень і технологій, а також представлено рішення проблеми. Основною технологією проекту є система штучного інтелекту, а точніше – згортова нейронна мережа. В якості основного алгоритму для виявлення номерних знаків використовується Mask R-CNN. Щоб представити обґрунтоване дослідження, систему протестували в різних середовищах (CPU, GPU, Raspberry PI 4) і на різних наборах даних.

Key words: automatic license plate recognition, machine learning, deep learning, convolutional neural networks, Mask R-CNN

Ключові слова: автоматичне розпізнавання номерних знаків, машинне навчання, глибоке навчання, згорткові нейронні мережі, Mask R-CNN

Introduction

License plate recognition systems (LPRS) have become a crucial technology in recent years, finding applications in various fields such as law enforcement, traffic management, and parking systems. LPRS plays a significant role in ensuring the safety and security of individuals and property, as well as facilitating efficient transportation management. One of the key challenges in LPRS is accurately detecting and recognizing license plates from images captured by cameras, which can be affected by various factors such as lighting conditions, vehicle orientation, and plate occlusion.



In recent years, deep learning-based methods, particularly those using convolutional neural networks (CNNs), have shown great promise in addressing these challenges. Among the many CNN architectures available, one particularly powerful and versatile option is the Mask R-CNN (Region-based Convolutional Neural Network) model. This model combines object detection and instance segmentation tasks in a single framework, allowing it to accurately identify and isolate different objects within an image.

Literature analysis

All ALPR systems can be classified as hardware [1] and software [2] systems. Software systems are used to perform analysis on obtained data (for example, on a server). Outdoor camera transmits a data stream and powerful computation server performs analysis of this stream. This approach allows installation of cameras in different environmental and weather conditions. Even if camera is getting broken, it can be easily and cheaply replaced.

Hardware systems uses another principle to perform analysis. Cameras are expensive and should be protected from several of weather conditions, but they perform on-site analysis (detection and extraction) of the license plate and distribute obtained information to a server. This approach has own benefits. For instance, because analysis are performed on-site it is allows manipulation of mechanisms without any delay. In addition, requirements to the server are significantly lower, as it should perform only persistence of the received data.

Firstly idea to create own ALPR system was based on software approach, but in order to understand the needs of the users and to test performance and capabilities of the system Raspberry PI 4 was chosen for hardware implementation. It is obvious that there are may be approach related problems with the process of recognition. Software implementations should be able to run multiple analysis separately to work with a bunch of incoming data flows. Hardware implementation should run as fast as possible in order to reach real-time detection.

Raspberry PI attracts interest of researches because of its simplicity and availability, that’s why so many different projects were made. One type of these projects is LPR projects. Different methods were discovered and implemented, but firstly it is important to understand capabilities of the platform (Table 1). Raspberry PI doesn’t have big computation power but still opens a wide field for experiments.

Table 1 – Technical characteristics of Raspberry PI 4

Processor	Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
RAM	4GB LPDDR4
Network and other communication	2.4 GHz and 5.0 GHz IEEE 802.11b/g/n/ac wireless LAN Bluetooth 5.0 Ethernet 2 × USB 3.0 ports, 2 × USB 2.0 ports
Storage	SD Card support
Power	5V DC USB-C

One of the methods is corner detection algorithm [3]. Using powerful open source computer vision library OpenCV different methods can be applied to perform extraction of the position of the number. The problem with such approach is low invariance capability. The slightest change in the color of the image may lead to crucial accuracy loss. Besides, no model can be trained, so all the situation should be processed manually at first.

Another way to implement hardware system is to use Raspberry PI as obtaining device and then perform correct analysis on the server [4]. Despite the good accuracy and speed, it is software based approach and not hardware based.

One of the possible hardware solutions is unmanned aerial vehicles. Concept of this solution will allow better adoption to changeable environment (Fig. 1).



Fig. 1 – Concept of ALPR drone

Despite the type of problems, related to implementation, there are also problems that all ALPR have in common. Camera position is the most important factor for accurate detection (Fig. 2). One of the goals of the work is to make the system efficient considering large camera shooting angles. Such restrictions complicate the use of modern ALPR systems in real situations on public roads, especially in the presence of multi-lane traffic, large camera shooting angles, bad visibility of license plates, and in difficult weather conditions [5, 6].

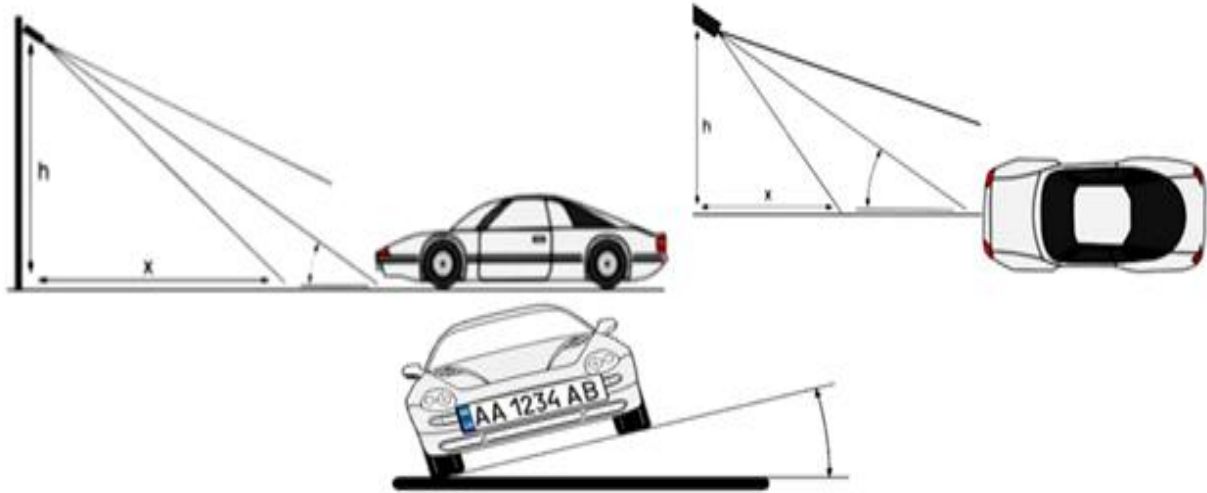


Fig. 2 – Camera position invariance

Different methods may come in handy when developing artificial intelligence system for license plate detection [7, 8, 9] Most of the researches refer to machine learning methods such as artificial neural networks. ANN is great tool to predict the behaviour of complex functions. In field of computer vision, convolutional neural networks are favourite [10, 11] (Fig. 3).

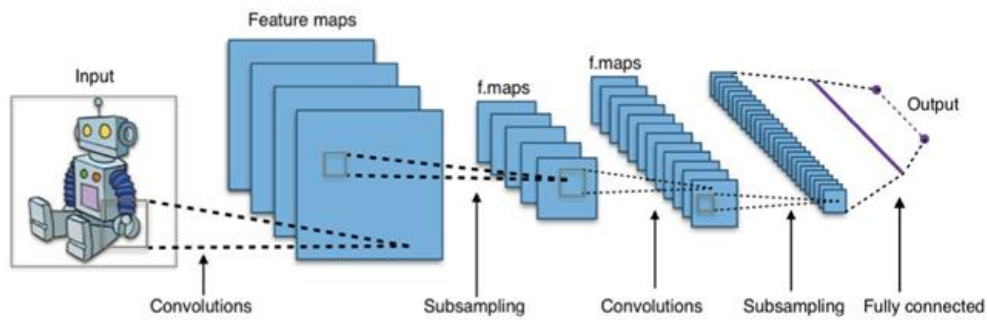


Fig. 3 – General convolutional neural network architecture

Semantic segmentation models are mostly based on object detection models. Mask R-CNN is one of them [12] (Fig. 4).

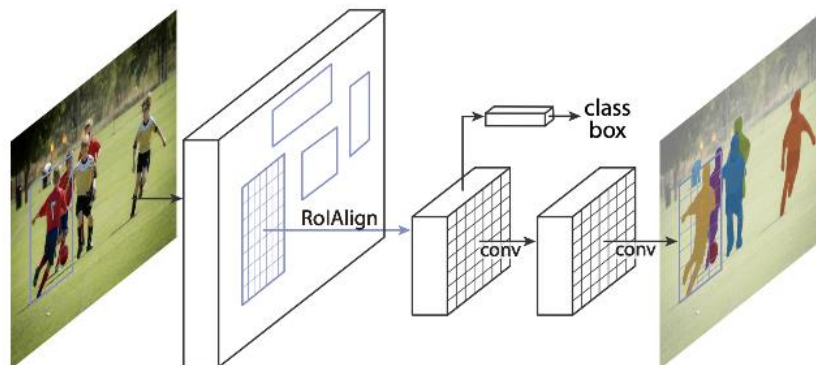


Fig. 4 – Mask R-CNN general architecture

Unique technic is used in order to process images. CNN (in our case ResNet101) performs feature map extraction. Then another ANN generates regions that are classified later. Fitting process of this network can be performed with the help of next loss functions:

$$L(u, p, t^u, v^p) = L_{cls}(p, u) + \lambda L_{loc}(t^u, v^p) + L_{mask}(m, u, p)$$



Where classification, localization and mask losses defined as:

$$L_{cls}(p, u) = -u \cdot \log p - (1 - u) \cdot \log(1 - p)$$

$$L_{loc}(t^u, v^p) = \sum_{i \in \{x, y, w, h\}} L_1^{smooth}(t_i^u - v_i^p)$$

$$L_{mask}(m, u, p) = -\frac{1}{m^2} \sum_{i=1}^m \sum_{j=1}^m [u_{ij} \cdot \log p_{ij} + (1 - u_{ij}) \log(1 - p_{ij})]$$

u – class of the object in particular region;

u_{ij} – class of the i, j pixel;

p – predicted class;

p_{ij} – predicted class of the i, j pixel;

v^p – real bounding box;

t^u – predicted bounding box;

λ – coefficient of the localization loss function weight (originally is 1);

$m \times m$ – size of the mask;

$$L_1^{smooth}(x) = \begin{cases} 0.5 \cdot x^2, & |x| < 1 \\ |x| - 0.5, & |x| \geq 1 \end{cases}$$

Results

For accuracy test another two license plate recognition systems were used: NomeroffNet [13] and SeeAuto (FF-Group) [14] (Fig. 5). To test models “Austria Number Plate Dataset” was used. Default accuracy metrics was used:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

TP – true positive (number plate exists and was correctly predicted);

TN – true negative (number plate doesn't exist and wasn't predicted);

FP – false positive (number plate doesn't exist but was predicted);

FN – false negative (number plate exists, but wasn't predicted).

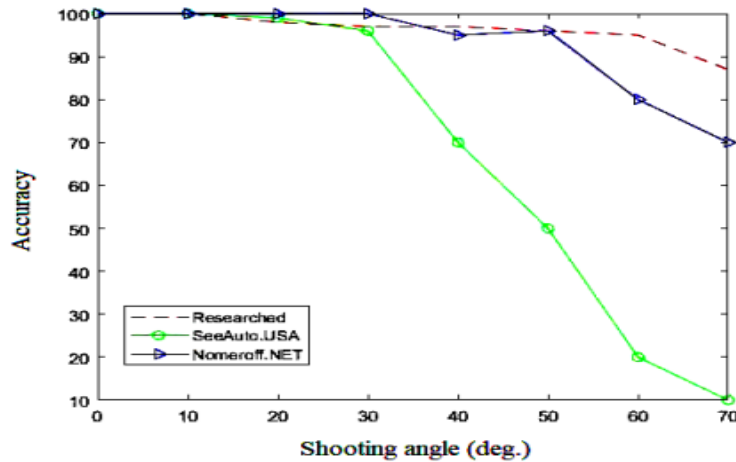


Fig. 5 – Accuracy comparison

To test another important metrics like precision, recall and F1-Score of the model Medialab LPR dataset was used [15] (Fig. 6).

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$



Fig. 6 – Difficult multi license plate situation

Totally, almost 1 thousand images were tested. The results are shown in Table 2.

Table 2 – Test metrics among datasets

	Metrics		
	Precision	Recall	F1-Score
Autoria Number Plate Dataset	0.96	0.95	0.95
Medialab LPR dataset	0.96	0.93	0.94

Raspberry PI is best platform for embedded systems research [16]. Large market of different modules makes this platform suitable for different projects. This platform allows testing of different potential hardware systems without any special knowledge or preparations.

In the beginning, two different cameras were considered: PI camera and default USB-Camera. To make the project more reproductive standard USB-Camera was chosen. Resolution of the camera is 640x480 pixels. Raspberry PI were connected to external monitor (Fig. 7) for results visualization, but it isn't necessary, because Raspberry PI has Wi-Fi and Ethernet interfaces for communication through net.



Fig. 7 – Raspberry PI connection scheme

Because of large modules like TensorFlow and OpenCV time required to process one image firstly was from 10 to 20 seconds. After installation of simpler TensorFlow Lite and OpenCV Lite results are better (around 5 to 15 seconds per image), but there is no way this system can be used in real-time. Significant optimization improvements must be made or platform computation capabilities must be greater.

Another way to apply developed system is drones. With the help of drones and great accuracy, it makes sense to use this system as part of the drone for automatic license plate detection in changing environment [17].



Although the system can't be applied for real-time number plate detection, the accuracy of the model doesn't depend on the platform, so the system shows same great accuracy results (Fig. 8).



Fig. 8 – License plate detection on Raspberry PI

As mentioned earlier it takes 5 to 15 seconds to process image with Raspberry PI.

First version of the system was deployed to server with next tech specification: Intel Xeon CPU E5-2630 v4 2.2 GHz. It takes up to 4 seconds per image to process.

Another performance test (main one) was performed on a laptop CPU Intel i7-9750H. Average time around 5 seconds was achieved. With modern GPU technologies the process takes a lot less time. GeForce GTX 1060 Ti shows result around 500 ms.

General conclusion

As the result of the research automated licence plate recognition system was build and tested in various of ways. LPR problem was studied to create accurate prediction system. A lot of information was processed in order to reach deep understanding of Mask R-CNN architecture and application possibilities.

As part of additional research current system was implemented with the help of Raspberry PI 4 for real-time tasks. Unfortunately, optimization improvements or more powerful platforms are needed to reach acceptable processing times delays. Great accuracy metrics show that the main goal of the project was reached. Studied system works better than compared in terms of large camera angles.

References

- [1]. Lubna, Mufti, N., Shah, S. A. A. (2021). Automatic Number Plate Recognition: A Detailed Survey of Relevant Algorithms. *Sensors*, 2021, vol. 21, iss. 9, article no. 3028. <https://doi.org/10.3390/s21093028>
- [2]. Li, H., Wang, P., Shen, C. (2019). Toward end-to-end car license plate detection and recognition with deep neural networks. *IEEE Transactions on Intelligent Transportation Systems*, 2019, vol. 20, no. 3, pp. 2351–2363. <https://doi.org/10.1109/TITS.2016.2639020>
- [3]. Aswinth, R. (2021). License Plate Recognition using Raspberry Pi and OpenCV. [Web resource]. URL: <https://circuitdigest.com/microcontroller-projects/license-plate-recognition-using-raspberry-pi-and-opencv> (last access 27.01.2023)
- [4]. Chiriac, R. L. (2020). I built a DIY license plate reader with a Raspberry Pi and machine learning. [Web resource]. URL: <https://towardsdatascience.com/i-built-a-diy-license-plate-reader-with-a-raspberry-pi-and-machine-learning-7e428d3c7401> (last access 27.01.2023)
- [5]. Psyllos, A., Anagnostopoulos, C. N., & Kayafas, E. (2011). Vehicle model recognition from frontal view image measurements. *Comput. Standards & Interfaces*, 2011, vol. 33, no. 2, pp. 142–151. <https://doi.org/10.1016/j.csi.2010.06.005>
- [6]. Kranthi, S., Pranathi, K., & Srisaila, A. (2011). Automatic number plate recognition. *International Journal of Advancements in Technology*, 2011, vol. 2, no. 3, pp. 408–422.
- [7]. Liubchenko, N., Nakonechnyi, O., Podorozhniak, A., & Siulieva, H. (2018). Automation of vehicle plate numbers identification on one-aspect images. *Advanced Information Systems*, 2018, vol. 2, no. 1, pp. 52-55. <https://doi.org/10.20998/2522-9052.2018.1.10>
- [8]. Podorozhniak, A., Liubchenko, N., & Heiko, H. (2020). Neural network system for license plates recognizing. *Control, Navigation and Communication Systems*, 2020, vol. 4, no. 62, pp. 88-91. <https://doi.org/10.26906/SUNZ.2020.4.088>



- [9]. Podorozhniak, A., Liubchenko, N., & Onishchenko, D. (2021). An intelligent system for recognizing vehicle number plates [Conference presentation]. I International science conference on multidisciplinary research, Berlin, Germany, 2021, January 19-21, pp. 1062-1066. <https://doi.org/10.46299/ISG.2021.II>
- [10]. Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86 (11), pp. 2278–2324. <https://doi.org/10.1109/5.726791>.
- [11]. Kuchuk, H., Podorozhniak, A., Liubchenko, N., & Onishchenko, D. (2021). System of license plate recognition considering large camera shooting angles. Radioelectronic and Computer Systems, 2021, no. 4 (100), pp. 82 – 91. <https://doi.org/10.32620/reks.2021.4.07>
- [12]. He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN / 2017 IEEE International Conference on Computer Vision (ICCV), Venice, October 22-29 2017. – Venice, 2017. – pp. 2980-2988. <https://doi.org/10.1109/ICCV.2017.322>
- [13]. Nomeroff Net. Automatic numberplate recognition system from AUTO.RIA.com. Version 0.2.3. [Web resource]. URL: <https://nomeroff.net.ua/> (last access 27.01.2023)
- [14]. FF-GROUP. [Web resource]. URL: <https://ff-group.org/seeauto> (last access 27.01.2023)
- [15]. Medialab LPR database. [Web resource]. URL: <http://www.medialab.ntua.gr/research/LPRdatabase.html> (last access 27.01.2023)
- [16]. Raspberry PI 4 Official site. [Web resource]. URL: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/> (last access 27.01.2023)
- [17]. Kaimkhani, N. A. K., Noman, M., Rahim, S., & Liaqat H. B. (2022). UAV with Vision to Recognize Vehicle Number Plates. Mobile Information System, 2022, vol. 4, article ID 7655833. <https://doi.org/10.1155/2022/7655833>

Отримана в редакції 28.06.2023. Прийнята до друку 02.08.2023. Received 28 June 2023. Approved 02 August 2023. Available in Internet 12 September 2023.

УДК 004.415.53:004.424

SURVEILLANCE AND SECURITY MODELS BASED ON THE ARDUINO UNO INTERFACE

Zakariashvili M.Z.¹, Sivsivadze G.², Chidrashvili M.³, Khechoshvili L.⁴

^{1,2,3,4} Iakob Gogebashvili Telavi State University, Telavi, Georgia

ORCID: <https://orcid.org/0000-0003-3903-6636>¹

E-mail: ¹mariam.zaqariashvili@tesau.edu.ge, ²guram.sivsivadze@tesau.edu.ge, ³mate.chidrashvili@tesau.edu.ge,

⁴luka.khechoshvili@tesau.edu.ge

Copyright © 2021 by author and the journal “Automation of technological and business – processes”.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0>



DOI: 10.15673/atbp.v15i3.2624

Abstract. The issue demonstrated in the article covers construction of surveillance and security systems on Arduino UNO platform Technological and Software aspect which are based on Quality Semiconductor Sensor-MQ135; PIR Motion Sensor and use of blocking peculiarities.

The project model integration process in both a virtual and physical context is described in detail and analyzed in the paper. The following are specifically used: An Arduino UNO microcontroller board, the Arduino Software IDE (Integrated Development Environment), a local code editor on the PC, and the virtual online platform tinkercad.com. The work is supplemented by pertinent electronic diagrams, schemes, software code, descriptions of fundamental functionalities, links to virtual project simulation, real-world application, and links to video recordings of the experiment in order to demonstrate the project model's smooth operation.

Анотація. Проблема, що розглядається в статті, охоплює побудову систем відеоспостереження та безпеки на платформі Arduino UNO. Технологічний та програмний аспект, які базуються на Quality Semiconductor Sensor-MQ135; PIR датчик руху та особливості використання блокування.

У статті детально описано та проаналізовано процес інтеграції моделі проекту як у віртуальному, так і у фізичному контексті. Конкретно використовуються: плата мікроконтролера Arduino UNO, Arduino Software