Vol. 6, No. 3, 2024

Vladyslav Vysotskyi[1], Nazariy Jaworski[2]

[1]Computer Design Systems Department, Lviv Polytechnic National University, S. Bandery street 12, Lviv, Ukraine, E-mail: vladyslav.m.vysotskyi@lpnu.ua, ORCID 0009-0005-9732-163X
[2]Computer Design Systems Department, Lviv Polytechnic National University, S. Bandery street 12, Lviv, Ukraine, E-mail: nazarii.b.yavorskyi@lpnu.ua, ORCID 0000-0003-1312-7302

SMART PARKING SYSTEM FOR LICENSE PLATE RECOGNITION BASED ON YOLO NEURAL NETWORK AND OPTICAL CHARACTER RECOGNITION

Received: November 04, 2024 / Revised: November 20, 2024 / Accepted: November 25, 2024

© Vysotskyi V., Jaworski N., 2024

https://doi.org/

Abstract. This paper describes a license plate recognition method, exemplified by training and deploying a machine learning model. The study uses the YOLO (“You Only Look Once”) neural network architecture and optical character recognition (OCR) techniques to extract license plate characters for real-time license plate recognition. Experimental tests, including model training, validation, and evaluation, demonstrate the effectiveness of these methods in enhancing automated access control in smart parking systems.

Keywords: Neural Network Models, YOLO, License Plates Recognition, Smart Parking, Optical Character Recognition

Introduction

The development of automated parking systems is becoming increasingly important as smart cities evolve and demand for efficient urban infrastructure grows. An important component of these systems is automated license plate recognition, which utilizes advanced computer vision and artificial intelligence (AI) technologies. The YOLO (“You Only Look Once”) [1] neural network architecture is notable for its real-time object detection capabilities, which can provide accurate and fast license plate identification.

The global adoption of AI-driven systems is expanding rapidly. At the same time, the smart parking market is also expected to reach $16.5 billion by 2030 [2]. This growth underscores both the ability and the importance of integrating robust machine learning models to optimize parking management and improve user experience. The ability to efficiently process data and provide results in real time makes machine learning an important tool in modern smart parking solutions.

Objectives and Problems of Research

This research has the following objectives:

1. preparing data and training a YOLO machine learning model for recognizing license plates of the Ukrainian standard;
2. studying and interpreting the results to investigate the quality of model training;
3. integrating the model with an existing OCR system and testing their operation;
4. determining the opportunities and directions for further research.

Main Material Presentation

Data preparation for further model training

In order to be able to detect a license plate and display its bounding box, the YOLO neural model must be trained beforehand. The training process requires a prepared dataset. This study utilizes a dataset of car images sourced from AutoRia [3]. This allows the use of images with Ukrainian license plates of various types. Thus, it almost completely covers the most important standard - the State Standard of Ukraine "DSTU" (Fig. 1).

The next step is to use the dataset to set the expected bounding boxes. This is required for the neural network training process itself. Training a YOLO neural network is a type of supervised learning which means that the model is trained on a large amount of data using appropriate labels. During training, the model receives input images and the corresponding coordinates of bounding boxes that define the objects to be recognized.

The Roboflow service (https://roboflow.com) can be used to prepare all types of datasets. Using its documentation [5], we upload our images (about 800 images selected in the previous step) and create a class to solve the classification problem. Fig. 2 illustrates this service in use.

Fig. 1. "DSTU" view of license plates

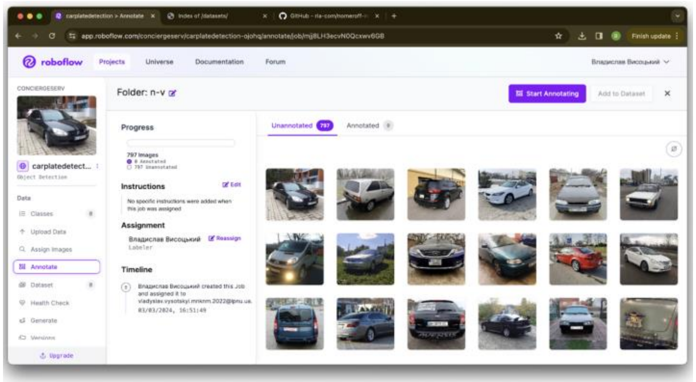

Fig. 2. Working with the Roboflow service

Fig. 3 shows how the bounding boxes are set by using the online editor provided by Roboflow.

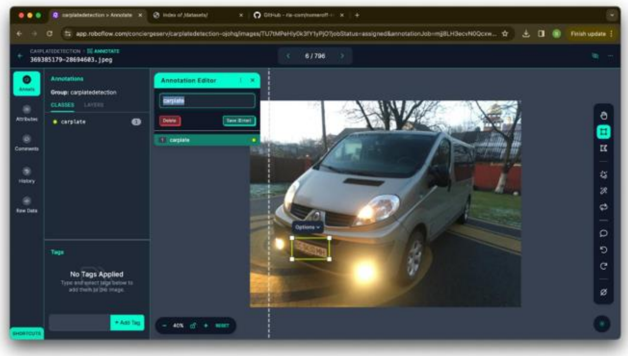

Fig. 3. Manual selection of bounding boxes for 800 images

After each image is manually annotated, the dataset should be split into images for training, testing, and validation. To balance the demands on the model, a proportion of 70% for training, 15% for validation, and 15% for testing was used (Fig. 4). The test set evaluates the final performance of the model on unknown data, suggesting generalizations; the validation set helps in tuning hyperparameters and prevents overfitting; and the training set provides most of the data needed to identify patterns. These ratios are often used because they offer enough information for each of the three steps without the risk of creating an imbalance in the model's estimation.

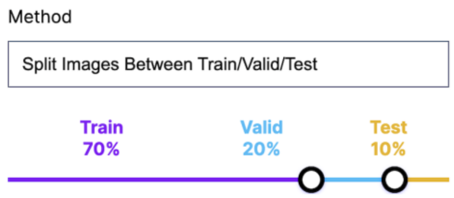

Fig. 4. Percentage breakdown of the dataset into training, validation, and test data

Training the YOLO network to detect a license plate in an image

After setting up the environment and downloading all YOLO *ultralytics* [4] and other required Python-dependencies, the training process can begin, as shown in Fig. 5. In order to start training, it is necessary to transfer data sets and parameters of how the model will be trained. The parameters used for training are 25 epochs with AdamW optimizer. This optimizer is often used for this type of task and dynamically adjusts the learning rate for each parameter, allowing the model to converge faster and more efficiently.

To accelerate the learning process on Nvidia graphic cards, it is crucial to ensure the proper installation and configuration of the CUDA driver, if supported by the hardware [6]. CUDA is a proprietary technology developed by Nvidia that allows GPUs to be used to perform various computing tasks much faster than can be done on a CPU. With thousands of relatively simple cores, GPUs enable simultaneous execution of many typical tasks. This speeds up the learning process significantly [7].

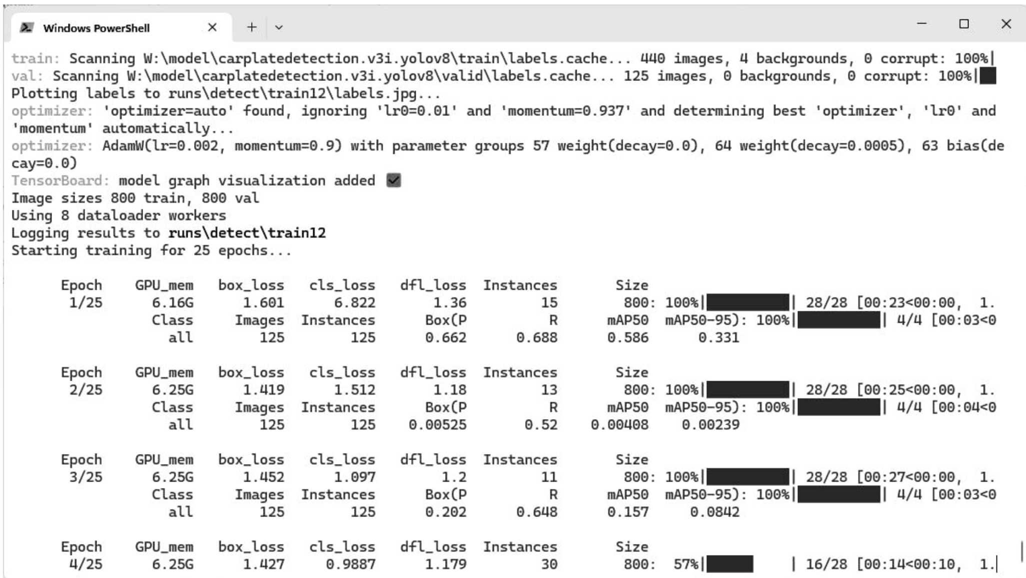

Fig. 5. Model training process

As a result of training, we obtain the weights (*weights/best.pt* file) and many different graphs. Let's look at a few graphs that allow checking the quality of the trained model.

Fig. 6 shows the precision curve depending on the level of confidence in the license plate recognition model.

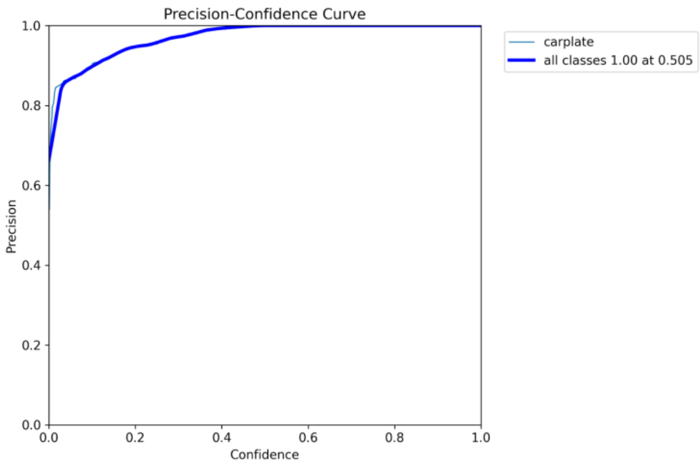

Fig. 6. Graph of precision-confidence curve

This graph shows that as the confidence level increases, the model's precision increases correspondingly, reaching a plateau at the highest level of 1.0. The initial increase in precision indicates that when higher confidence thresholds are chosen, the model makes fewer errors, or in other words, produces fewer false positives. This is typical for systems where low confidence thresholds make it easier to classify objects as positive, which leads to a higher false positive rate.

To better understand the concept of accuracy, it can be expressed through the equation Eq. 1:

$$P = \frac{TP}{TP + FP}, \tag{1}$$

where *TP* are true positives and *FP* are false positives.

The fact that precision remains high even at high confidence levels indicates that when the model classifies an object, it has a high probability of being correct. This may be a sign that the model is good at distinguishing between classes of objects.

However, it is important to remember that high precision alone does not guarantee that a model is effective. For example, if a model has a high accuracy but a very low confidence factor, it may mean that it is very careful in selecting positive cases for classification, but misses many true positive results. The optimal balance between precision and confidence depends on the specific goals and requirements of the task.

The second important graph (Fig. 7) shows the curve of the dependence of recall on confidence. The recall remains at 1.0 for most confidence levels, which means that the model detects all real positive cases before the confidence threshold becomes too high. A sharp drop in recall occurs only at very high confidence thresholds, which is expected as the model starts to miss some true positive cases.

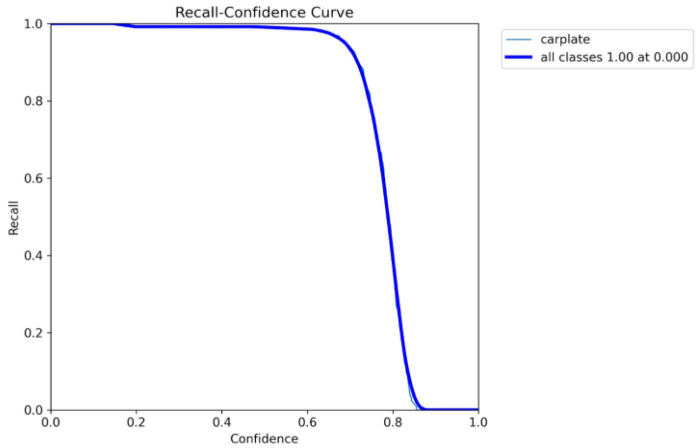

Fig. 7. Graph of recall-confidence curve

Based on the presented curve plots, we can conclude that the model demonstrates high accuracy and recognition ability at a low confidence threshold. This indicates its ability to effectively classify true positive cases. Since both curves reach high values at a low confidence threshold and maintain these values when the threshold is increased, we can say that the model strikes a good balance between preventing false positives and missing true positives, making it well balanced for practical applications.

The next image generated by YOLO is the results of the model (Fig. 8), presented in the form of bounding boxes with the percentage of confidence that the image area belongs to a certain class. These results were collected during the model validation phase and, as can be seen, the confidence levels are above 0.8 (where 1 is the maximum value and 0 is the minimum value). It can be visually confirmed that the model correctly identified the frames in the images from the test set. Thus, we can be sure that the model has been trained and is ready for further use. As a result of the training, we received the file *weights/best.pt*, which contains the weights for the model. The file size is relatively small and is just 22 MB.

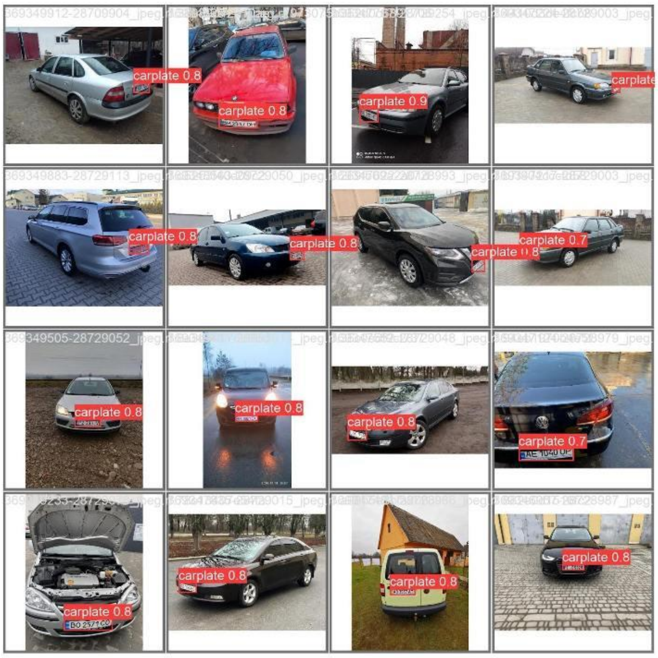

Fig. 8. Verification of the model on test data

An important concept of YOLO networks (and other detector networks) is Non-Max Suppression [8] (NMS). NMS is a post-processing stage that allows selecting only one bounding box among many boxes corresponding to the same object. The principle of operation is shown in Fig. 9. The Non-Max Suppression algorithm takes many frames with different levels of "confidence" and, using the Jaccard index ("Intersection over Union") and a certain numerical threshold, produces a single frame that best represents the union of all previously obtained bounding boxes. This post-processing should be enabled, as it is extremely important for further correct operation of OCR, where the extracted frame will be transferred to.

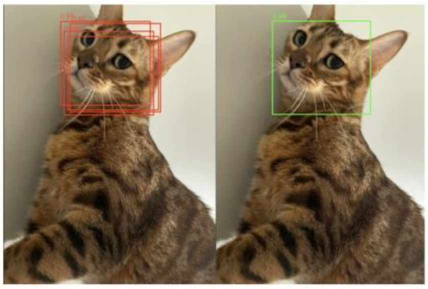

Fig. 9. Non-Max Suppression selects one merged frame of class "Cat" out of many

Results and Discussion

As a result, an application that uses the camera module cyclically reads information from the sensor frame by frame was developed. Fig. 10 illustrates the system in IDLE mode, processing the video stream at a rate of 12 FPS. Each frame is immediately transmitted to the YOLO neural model, which finds the bounding box for the license plate, if it is present. The license plate images obtained in this way are sent to the OCR, which extracts alphanumeric characters from the images.

For this work, a simple but effective recognition library called *EasyOCR* was used. This library [9] is well-known for its ease of use and support for many languages, including Cyrillic. However, in scenarios with high load and real-time processing, its speed can be a bottleneck in the system, especially when processing high-resolution video or in difficult conditions such as poor lighting or distorted license plates.

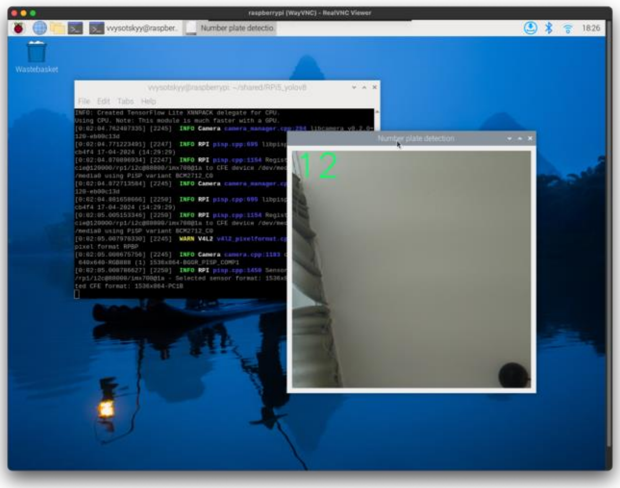

Fig. 10. IDLE mode

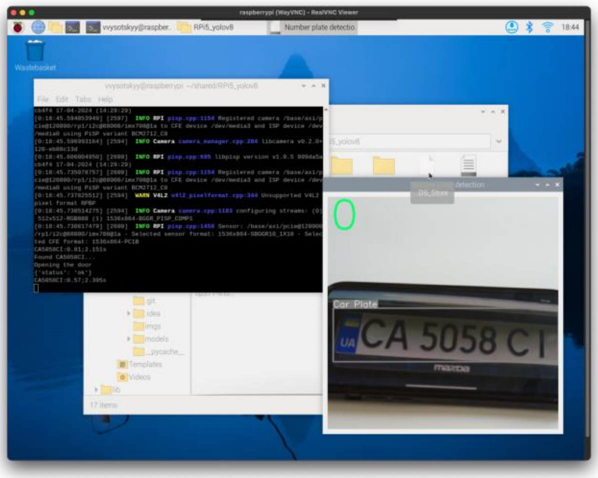

Fig. 11. Number recognized

The recognized license plate is shown in Fig. 11. OCR returns results with a certain level of confidence, and only results exceeding a predefined confidence threshold are considered.

As we can see, the trained neural network was able to extract the image with the license plate coordinates, which was then processed using OCR. Future improvements within this study may include, but are not limited to, optimizations such as parallelizing the detection and recognition processes, using more efficient image processing algorithms, improving OCR performance through the potential use of machine learning models, including transformer-based models, using newer versions of YOLO with new State of the Art tools, switching to LiteRT [10] (formerly TensorFlow Lite), and others.

Conclusions

This paper discusses a license plate recognition method for modern smart parking systems. This solution is based on the YOLO neural model for license plate recognition and includes OCR system for character recognition.

During the study, a machine learning model based on the YOLO architecture was trained to recognize license plates in real time and integrated with a recognition system to extract alphanumeric characters.

The main theses arising from the results of this study are as follows:

- the development of AI models and methods for license plate recognition can be an effective solution for automating access control in smart parking systems;

- the development and implementation of effective computer vision systems, including object detection and character recognition, require in-depth knowledge of machine learning and optimization methods;

- existing solutions for developing, training, accelerating, and optimizing neural models are constantly evolving, enabling more efficient and cost-effective deployments.

Next steps may include optimizing the current convolutional neural network and OCR. In particular, the YOLO neural model can be tweaked by switching to LiteRT (formerly TensorFlow Lite), using its quantization and inference acceleration features. This will provide a real-time performance boost without significantly reducing accuracy, and will also allow the trained network to run on low-end devices, making the Smart Parking solution that may be developed more cost-effective.

Other improvements mentioned here, such as parallelizing the detection and recognition processes using available Python or other programming language tools, using more efficient algorithms, and keeping the utilized architecture up to the latest version, can also be considered as potential directions for the development of this research.

In addition, the possibility of deeper integration of machine learning models for broader recognition tasks, such as recognizing not only license plates but also vehicle make and model, is to be explored in the future. This will enhance the system's ability to provide complex data, which will increase its impact on smart parking solutions.

References

[1] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 779-788, https://doi.org/10.48550/arXiv.1506.02640

[2] *[Electronic resource]* Smart Parking Market Size is projected to reach USD 16.54 Billion by 2030, growing at a CAGR of 13.6%: Straits Research,https://www.globenewswire.com/en/news-release/2023/09/14/2743480/0/en/Smart-Parking-Market-Size-is-projected-to-reach-USD-16-54-Billion-by-2030-growing-at-a-CAGR-of-13-6-Straits-Research.html

[3] *[Electronic resource]* AutoRia Dataset, https://nomeroff.net.ua/datasets

[4] *[Electronic resource]* Ultralytics YOLOv8 Docs, https://docs.ultralytics.com/models/yolov8

[5] *[Electronic resource]* Roboflow Documentation, https://docs.roboflow.com

[6] *[Electronic resource]* CUDA Toolkit Documentation, https://docs.nvidia.com/cuda/

[7] Afif, Mouna & Said, Yahia & Atri, Mohamed. (2020). Computer vision algorithms acceleration using graphic processors NVIDIA CUDA. Cluster Computing. 23. 10.1007/s10586-020-03090-6.

[8] *[Electronic resource]* Sambasivarao K, “Non-maximum Suppression (NMS), A technique to filter the predictions of object detectors.”, Towards Data Science, https://towardsdatascience.com/non-maximum-suppression-nms-93ce178e177c

[9] *[Electronic resource]* EasyOCR Github page, https://github.com/JaidedAI/EasyOCR

[10] *[Electronic resource]* LiteRT overview, https://ai.google.dev/edge/litert

Владислав Висоцький[1], Назарій Яворський[2]

[1]Кафедра систем автоматизованого проектування, Національний університет Львівська політехніка, вул. С. Бандери 12, Львів, Україна, E-mail: vladyslav.m.vysotskyi@lpnu.ua, ORCID 0009-0005-9732-163X

[2] Кафедра систем автоматизованого проектування, Національний університет Львівська політехніка, вул. С. Бандери 12, Львів, Україна, E-mail: nazarii.b.yavorskyi@lpnu.ua, ORCID 0000-0003-1312-7302

СИСТЕМА РОЗУМНОГО ПАРКУВАННЯ ДЛЯ РОЗПІЗНАВАННЯ НОМЕРНИХ ЗНАКІВ НА ОСНОВІ НЕЙРОМЕРЕЖІ YOLO ТА ОПТИЧНОГО РОЗПІЗНАВАННЯ СИМВОЛІВ

Отримано: Листопад 04, 2024 / Переглянуто: Листопад 20, 2024 / Прийнято: Листопад 25, 2024

© Висоцький В., Яворський Н., 2024

Анотація. У статті описано метод розпізнавання номерних знаків на прикладі навчання та розгортання моделі машинного навчання. У дослідженні використовується архітектура нейронної мережі YOLO («You Only Look Once») і методи оптичного розпізнавання символів (OCR) для вилучення символів номерних знаків для розпізнавання номерних знаків у реальному часі. Експериментальні випробування, включаючи навчання моделі, валідацію та оцінку, демонструють ефективність цих методів у покращенні автоматизованого контролю доступу в розумних системах паркування.

Ключові слова: Моделі нейронних мереж, YOLO, розпізнавання номерних знаків, розумне паркування, оптичне розпізнавання символів