An Efficient real-time Moroccan automatic license plate recognition system based on the YOLO object detector

Zainab Ouardirhi[1,2], Sidi Ahmed Mahmoudi[2], Mostapha Zbakh[1], Mohamed El Ghmary[3], Mohammed Benjelloun[2], Hamd Ait Abdelali[4], and Hatim Derrouz[4]

[1] Communication Networks Department, ENSIAS, Mohammed V University in Rabat, Morocco
[2] Computer and Management Engineering Department, Polytechnic Faculty in Mons (FPMS-UMONS),Mons,Belgium
[3] Department of Computer Science, FSDM, Sidi Mohamed Ben Abdellah University, Fez, Morocco
[4] Microelectronics Embedded System department, Moroccan Foundation for Advanced Science Innovation and Research, Rabat, Morocco

Abstract. In this paper, we adopt a powerful deep neural network (DNN) to tackle the problems of detecting car license plates (LPs) and recognizing Arabic letters in natural scene images. Our Automatic License Plate Recognition (ALPR) system is built using state-of-the-art methodologies and techniques to provide the optimal speed/accuracy trade-off at each level. Images of vehicles and Moroccan and European LPs were used to train the neural networks. The system was able to correctly detect and recognize in the test set all the characters existing in a LP with an accuracy of 91.11% and 98.89% taking into account that only one character is misread. Using a high-end GPU, our system likewise produced outstanding real-time execution results.

Keywords: Deep Learning, Image processing, CNN, YOLO, Zone of interest, Detection, Recognition, Voting system, License Plates.

1 Introduction

Modern civilization has seen a huge growth in intelligent systems (IS) for the purpose of processing information. The combination of internet of things (IoT) and artificial intelligence (AI) technologies has allowed us to create intelligent information processing systems at an industrial level. However one of the difficult parts of developing and implementing such industrial IS is data collection.

Vehicles have now become one of the most frequently regarded as conceptual resources in information systems, as a result of the widespread adoption of information technology in various parts of modern life. Information from reality must be transformed into autonomous information systems. This can be done by human agents or particular intelligence equipment. Among these intelligent devices, we must mention Automatic License Plate Recognition (ALPR) system.

ALPR technology is growing in use, especially in Surveillance and Intelligent transportation systems, with relevant applications in several sectors such as command force, road safety, parking management [1], traffic flow control and toll collection [2].

Current ALPR systems achieve exemplary performance in controlled environments; however, performance is degraded when processing complex scenes. These devices can deal with a variety of constraints and uncontrollable conditions, such as particular detectors or viewing angles, uneven illumination, adequate lighting requirements, image blurring, capture in a predetermined zone, occlusions, and so on. Deep Learning (DL) methods have emerged as a useful element in the present sector. Indeed, with the advent of DL techniques, the accuracy of many pattern recognition tasks has improved.

In computer vision, many methods of achieving peak performance explore a kind of Convolutional Neural Network (CNN). The problem with these techniques is that they typically require tons of data to be trained from scratch, although most of the existing methods explore Transfer Learning to alleviate this problem. In addition, the more data you have, the deeper your network can be, allowing increasingly complex patterns to be recognized.

In this paper, we propose an end-to-end, efficient and layout-independent ALPR system using a YOLO-based model at all stages. In order to obtain the best speed-accuracy trade-off, the entire system was trained on a total of 25 510 annotated images that consists of vehicles and Moroccan and European LPs. We will start by reviewing related studies along with their limitations in section 2. Presenting the proposed system in section 3. Finally, before moving on to the conclusions, describe the experimental setup in section 4.

2 Related Work

2.1 ALPR systems

This section explores recent ALPR research that employs deep learning models. We'll go over the relevant literature from prior work in the areas of license plate detection (LPD) and license plate recognition (LPR), as well as their limitations.

The authors of [3] have suggested a real-time, end-to-end ALPR system based on deep convolutional neural networks (CNNs). In order to get better results, the method seeks to identify the front and back views of automobiles and LPs that are functioning in a cascaded mode. After that, characters within a cropped LP region are detected and recognized.

In [4], the authors suggested an ALPR system that was focused on locating and reading LPs in challenging situations. For LP detection and character segmentation, a CNN-based model called You Only Look Once version 3 (YOLOv3) was used, and a Convolutional Recurrent Neural Network (CRNN) [5] for character recognition (CR).

A study made in [6] presented a new system for the recognition of multinational LPs with a three-layer architecture, including LPD, Unified Character

Recognition (UCR), and Multinational License Plate Layout Detection. In this approach, the authors used a more simplified YOLOv3 architecture for LPD and another YOLOv3-SPP with Spatial Pyramid Pooling block for CR stages.

Authors in [7] proposed a combination of YOLO and sliding-window processes. In their method, each character of Taiwan's vehicle LPs is detected by the sliding-window, the YOLO framework then identifies each class of object, plate, digit, and letter in the window.

In [8], authors used specific CNNs for each ALPR stage. The models used are: Fast-YOLO [9], YOLOv2 and CR-NET [10], an architecture inspired by Fast-YOLO for character segmentation and recognition.

It was proposed in [11] a novel LP detection and character recognition algorithm based on a combined feature extraction model and Backpropagation Neural Network (BPNN) which is adaptablein weak illumination and complicated backgrounds.

The authors of [12] have developed an ALPR system that can be used in a mobile application to recognize Egyptian LPs. The algorithm aims to first apply preprocessing on the acquired image, character detection using segmentation, and then character recognition by extracting features using Speeded Up Robust Features (SURF) [13].

In [14], the authors have proposed a method for Chinese vehicle LPR, which employs YOLOv2 detector for LPD and a CRNN for CR stages. Their recognition architecture consists of a CNN for context feature extraction, and a two-layered Gated Recurrent Unit (GRU) [15] for feature sequences decode.

In [16], authors used a cascade structure to read the LP. The model first detects the character region using a CNN classifier, employed in a sliding window fashion across the entire image, to generate bounding boxes independently at each scale using the run Length Smoothing Algorithm (RLSA) and Connected Component Analysis (CCA). Then, the generated boxes are filtered by geometric constraints and refined by the edge feature of LP. And finally, another CNN classifier was used to verify the remaining bounding box.

Authors of [17] used a cascaded framework composed of a fast region proposal network and a R-CNN network to extract LP. The model aims first to generate the LP candidates using a light-weight RPN network. Then extracts the Region of Interest (ROIs) from the original image using the sampler. And finally, using the R-CNN network, the model classifies the candidate plate and regresses four corners of the LP.

The authors in [18] developed a lightweight ALPRmodel that may be implemented entirely on embedded devices such as the Raspberry Pi3. To achieve the lowest memory consumption for the detection stage, they employed a mix of a MobileNet [19] feature extractor with fewer parameters and a Single Shot Detection model. They also used LPRNet20 for character recognition, which is a powerful but computationally economical network.

In [20], the authors developed a CNN-based method called MD-YOLO, inspired by the YOLO framework, to realize multi-directional car LPD. They proposed the angle deviation penalty factor (ADPF) to approximate the intersection ratio between predicted value and tag value. And they chose leaky and identity functions as activation functions rather than ReLU function in order to identify negative rotation angle values. A prepositive CNN attention model called ALMD-YOLO was employed prior to the implementation of MD-YOLO considering that the LP is usually very small.

2.2 Limitations

Most of the related work in the ALPR context that we discussed section 2 has common limitations, as for instance the previous works that are dealing with multinational LPs were trained and tested on datasets from various countries, however, Moroccan LPs were never included. Certain previous works used many techniques, which results in an increase in the execution time of the ALPR system. Some of them also faced on the training faze the limitation of having smaller memory and computational power, and lack of training dataset.

3 Proposed system

In this section, we describe our proposed methodology to detect and recognize LPs. The input to the system is a video of a moving vehicle and the output is the textual form of that vehicle's LP content, as shown in figure 1.

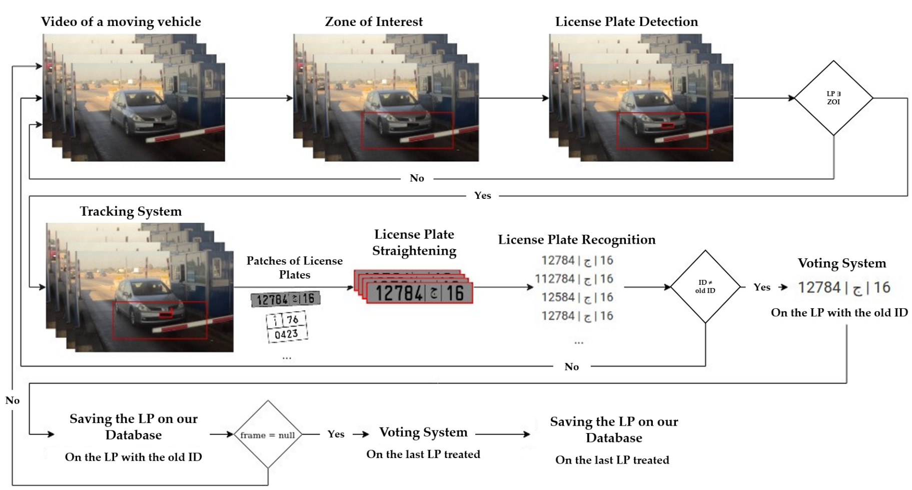

Fig. 1. Our proposed ALPR system block diagram : Following the acquisition of the input video, the system first detects the license plate when it appears in the manually defined zone of interest. For the period that this license plate is in this zone, it is tracked, extracted, and straightened in order to best recognize its content. Following the recognition phase, the system then employs a voting system that filters the most frequent license plate content in each image and saves it in our database for later use.

3.1 Zone of Interest

We created a Zone of Interest (ZOI) to allow the system to focus on one LP at a time, obtain a single good detection result for that LP, and only apply the recognition phase to plates with clear and easily recognizable characters. The next steps will be followed until the LP is validated to exist in that ZOI, as shown in figure 3.1.

Fig. 2. Before and after accessing the Zone of Interest.

3.2 License Plate Detection

Since LPs in our case tend to appear in scenes in small sizes, we devoted our research to find a model capable of detecting and recognizing an object of any size in an image. YOLOv3 [21] (figure 3.2), most well and reliable version of the YOLO model created by Joseph Redmon and his collaborators, is the most common deep object detector in practical applications. Indeed, it can detect small and overlapping compact objects. This YOLO version is slightly bigger than YOLOv2 [22], but it is more accurate and can detect faster.

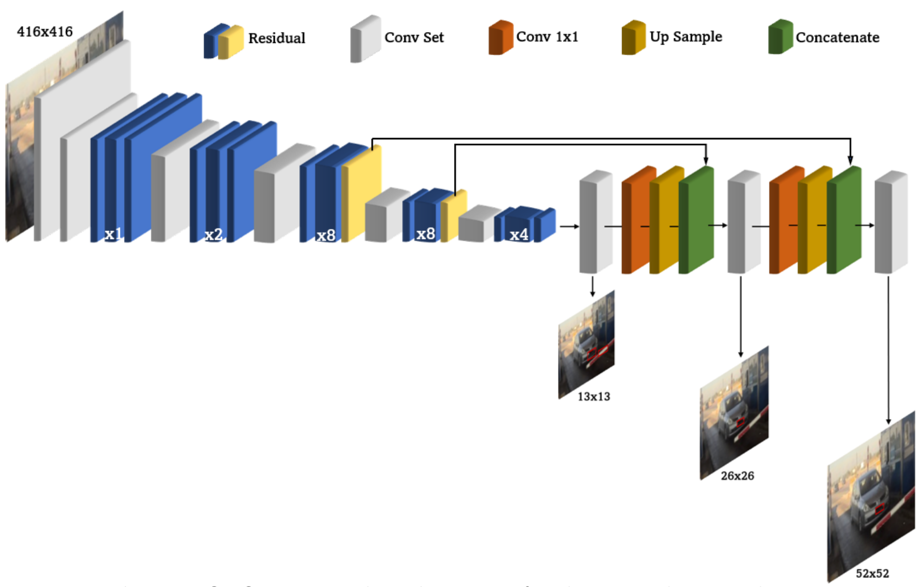

Fig. 3. YOLOv3 network architecture for detecting license plates.

Ultimately, we employed YOLOv3 network to detect LPs in order to achieve a decent balance between the accuracy rate and the execution duration. On a powerful computer, this network was established (and trained) to accomplish real-time detection. Darknet-53 is used by YOLOv3 as a feature extractor. It is still more efficient than ResNet-101 or ResNet-152 [23] and far more potent than the Darknet-19 used by YOLOv2.

Therefore, to use YOLOv3, we need to change the last number of layers taking into account the number of classes and the number of filters, we set them respectively to 18 and 1 where the class in our case is the LP (the last 3 layers of the network are shown in table1.). The number of filters was calculated based on the Eq (1), where B is the predicting boxes and C is the number of classes we want the model to detect.

$$N\_Filters = (B \times (5 + C)) \tag{1}$$

Table 1. The last three layers of YOLOv3's feature extractor (Darknet-53) for LPD phase

Layer	Type	Filters	Size	Input	Output
...	...	...	...	...	...
103	conv	128	1x1/1	52x52x256	52x52x128
104	conv	256	3x3/1	52x52x128	52x52x256
105	conv	**18**	1x1/1	52x52x256	52x52x18
106	detection				

3.3 Tracking System

Many computer vision and pattern recognition applications, including surveillance, vehicle navigation, and autonomous robot navigation, depend on object detection and tracking, also known as Object Tracking. Any tracking technique actually needs an object detection mechanism, whether it is in each frame or when the item first shows in the movie.

Additionally, object detection is technically unable to discriminate between several frames of the same object in an image. This has a negative impact on LP recognition and ultimately results in duplicate data. In order to overcome this problem, we will be tracking our LPs, so that each one of them becomes unique, as shown in figure 4.

The challenge of roughly tracking an object's trajectory in the picture plane as it travels across a scene is known as object tracking. Its goal is to find an object's position in each frame of the video in order to construct the path for that object over time.

The Tracking system we used, works well with occlusion, it employs a Hungarian algorithm [24] that can determine if an object in the current frame is

Fig. 4. Before and after applying the tracking system.

identical to one in the previous frame. It will be used for id attribution and association. The system makes also use of a Kalman Filter [25], which is a method for predicting future locations based on present ones. It may also estimate current position more accurately than the sensor or algorithm. It will be used to have better association.

3.4 License Plate Straightening Algorithm

Our LP does not automatically straighten after the detection and tracking phases, which makes the recognition phase ineffective. As shown in figure 5, we developed a license plate straightening (LPSt) algorithm to enhance the visibility of our discovered LP.

The algorithm works as follows: first, we obtain the image of the LP that needs to be straightened; next, we use the Canny Edge Detector to identify the edges of each object in the image; next, we use the Hough Line Transform to draw the longest lines, which in this case are the LP's edges; and finally, we rotate the resulting image in relation to the angle determined by two points that exist on the lines that we chose in the previous step.

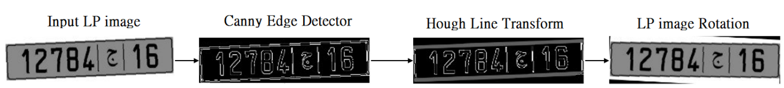

Fig. 5. Steps we followed to straighten the detected LP.

3.5 License Plate Recognition

We used YOLOv3 for the character recognition phase since as we explained in section 3.2 it can detect small objects and accept input images of various sizes. Similarly to the LPD phase, we set the number of filters to 237 and the number of classes to 74, which represent the number of Arabic and French letters, separators and numbers.

3.6 Voting System

Each time an LP enters the ZOI, the system detects it and recognizes its content, as discussed in the preceding subsections. Assuming that this process is applied to the same LP multiple times and that inaccurate recognition results are occasionally obtained, our system saves the position and content of each LP that appears at the level of each frame of the input video, so that it can later select the most frequent LP content in each image and save it in our database for future use. As illustrated in figure 6, not only is content redundancy at the database level eliminated, but our results are also more efficient and accurate.

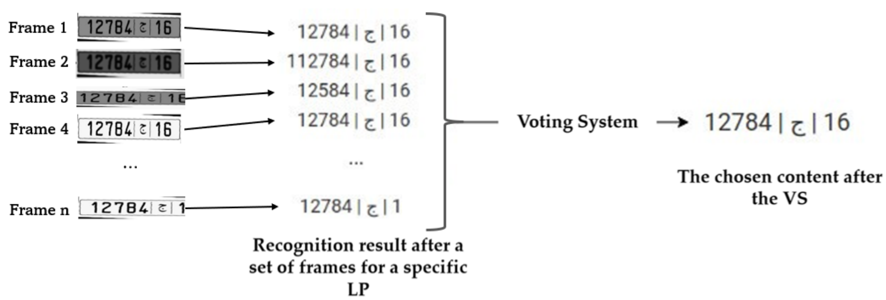

Fig. 6. Before and after the license plate voting system.

4 Experimental Results

In this section, we conduct the experiments to verify the effectiveness of the proposed system. Using a confidential database given by the MAScIR foundation, we trained our model with a Batch size of 64 and a Subdivision of 16 using 7 500 full resolution and high definition 1600x1200 size images of different types of vehicles for LPD, and 9 250 manually annotated Moroccan LPs and 8 760 manually annotated European LPs 400x100 size images for LPR, on an NVIDIA GTX 1050 Ti GPU (16GB of memory and 960 CUDA10.2 cores), a machine with an Intel 6core processor and a frequency of 3.5GHz per core under the Linux environment. We used 70% of our annotated data set for training and the rest for validation.

4.1 License Plate Detection

In order to know the performance of the YOLOv3 license plate detection model, we measured its relevance by calculating the precision, the recall and then conclude the F1 score using a confusion matrix which is built by putting respectively on the rows and columns of benchmark data which in our case are 1400 positive images containing vehicles, and 1400 negative images containing random objects, which were randomly selected from the internet.

According to the test performed on the system using sets of positive and negative images, we managed to obtain the confusion matrix presented in table 2. Where images containing LPs and the system correctly detecting them are true positives (tp), images containing LPs and the system incorrectly detecting

Table 2. Confusion matrix of the LPD phase

	Positive condition	Negative condition
Positive predicted	**tp** = 1247	**fp** = 53
Negative predicted	**fn** = 100	**tn** = 1400

them are false positives (*fp*), images without LPs and the system did not detect anything are true negatives (*tn*), and images that don't contain LPs but the system has detected objects are false negatives (*fn*), same for images that do contain LPs but the system has not detected objects.

Table 3. Evaluation of the LPD phase.

Method \ Metric	Accuracy	Recall	F1 Score
YOLOv3 [21]	66%	91%	76%
YOLO-LPD	95%	93%	94%

Based on the confusion matrix in table 2, we can extract the table 3 that concludes that our LPD system exhibits a very high level of accuracy. We were able to outperform the original YOLO model [21] with the help of our enhanced version (YOLO-LPD). In less than 680 ms and with a 94% F1 score, the YOLO-LPD network was able to detect one class in a 1600x1200 image.

4.2 License plate recognition

In order to know the performance of this model in this phase we tested our LPR on 2443 images of Moroccan and European LPs, and we obtained the results presented in the table 4.

Table 4. Evaluation of the LPR phase.

Total number of images	Entirely correct	>=1 CR error
2443	2336 (**95.62 %**)	107 (**4.38 %**)

This table shows that the model recognized 95.62% of the images well, which means our license plate recognition algorithm is very accurate.

4.3 Quantitative Comparison

In this section we report a comparison in order to show the influence of the proposed strategies on the accuracy rates of our proposed ALPR system. The results from testing the model without any techniques (YOLO-no LPSt and VS), with the LP straining technique (YOLO-LPSt), with the voting system (YOLO-VS), and finally with all the approaches combined, which represents our final Moroccan Automatic License Plates Recognition (MALPR) system, are shown in table 5.

Table 5. The influence of the proposed strategies on the accuracy rates of our MALPR system.

Method \ Metric	Accuracy	Recall	F1 Score
YOLO-no LPSt and VS [21]	67%	94%	78%
YOLO-LPSt	82%	97%	89%
YOLO-VS	89%	**98%**	93%
MALPR	**97%**	**98%**	**97%**

This table makes the effects of the strategies we suggested very evident, and it shows how much more accurate our suggested MALPR system is. This just helps to illustrate how well our model handles the challenge of multi-directional car license plate detection and recognition as well as the removal of redundant data.

4.4 System Assessment

To evaluate the entire system, we calculated the accuracy for the two levels: the LPD level and the LPR level. For the first level, we calculated the percentage of the LPs detected, then for the second level, we calculated the percentage of LPs when all characters were misclassified, the percentage of LPs when a single character was misclassified, the percentage of LPs when only two characters were misclassified, and the percentage of LPs when more than two characters were misclassified.

We tested the system on 457 positive images containing vehicles. table 6 shows the result of calculating metrics for the license plate detection part: recall and accuracy. The accuracy calculation results for the license plate recognition part are summarized in table 7.

Table 6. Final output evaluation. LPD phase test results.

Number of positive images (A)	Number of LPs correctly detected (B)	False positives (C)	Recall (Bx100/A)	Precision (Bx100)/(B+C)
457	**450**	**7**	**98.47 %**	**98.47 %**

Table 7. Final output evaluation. LPR phase. Accuracy considering: fully corrected LPs, at least 1 wrong characters, at least 2 wrong characters and more then 2 wrong characters.

LP detected	Good recognition	1 character badly detected	2 characters badly detected	>2 characters badly detected
450	**410 (91.11 %)**	**35 (7.78 %)**	**5 (1.11 %)**	**0 (0 %)**

Conclusion

This paper presented an end-to-end ALPR system based on Deep Learning. The main objective for building such a system was to meet the needs based on content indexing, content analysis of images and videos, and recognition of LPs.

The development of the proposed system was mainly based on the use of the YOLO model on the two main subtasks of the system, namely the detection and recognition of the license plate.

We have performed several tips and techniques on the system to improve recognition results, such as defining an ZOI to speed up the process by working on one LP at a time, applying a few modifications on the detected LP to work with clear and easy to recognize characters, and finally run a voting system on each LP, mainly to avoid redundancies when adding it to our database.

This strategy was essential to obtain exceptional results because we avoided errors of often misclassified characters and also a remarkable number of predicted characters. This was proven by the tests we performed in the last section on our dataset. Our system was able to achieve a detection rate of 98.47% and a complete recognition rate of 91.11%. These results are satisfactory for some real-world automatic license plate recognition applications.

As future work, we intend to explore new CNN architectures to further optimize (in terms of speed and accuracy) the ALPR system; We also want to include vehicle manufacturer and model recognition in the system pipeline so that our dataset provides such information; Make the system capable of recognizing multinational LPs by annotating multinational LPs and characters; Improve the results obtained by analyzing cases of failure; Finally, it would be interesting to integrate this license plate recognition system into a real-time on-board system.

References

1. T. Sirithinaphong and K. Chamnongthai, "The recognition of car license plate for automatic parking system," in *ISSPA'99. Proceedings of the Fifth International Symposium on Signal Processing and its Applications (IEEE Cat. No. 99EX359)*, vol. 1, pp. 455–457, IEEE, 1999.
2. P. Davies, N. Emmott, and N. Ayland, "License plate recognition technology for toll violation enforcement," in *IEE Colloquium on Image Analysis for Transport Applications*, pp. 7–1, IET, 1990.
3. S. M. Silva and C. R. Jung, "Real-time license plate detection and recognition using deep convolutional neural networks," *Journal of Visual Communication and Image Representation*, vol. 71, p. 102773, 2020.

4. W. Riaz, A. Azeem, G. Chenqiang, Z. Yuxi, W. Khalid, *et al.*, "Yolo based recognition method for automatic license plate recognition," in *2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA)*, pp. 87–90, IEEE, 2020.
5. B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 11, pp. 2298–2304, 2016.
6. C. Henry, S. Y. Ahn, and S.-W. Lee, "Multinational license plate recognition using generalized character sequence detection," *IEEE Access*, vol. 8, pp. 35185–35199, 2020.
7. R.-C. Chen *et al.*, "Automatic license plate recognition via sliding-window darknet-yolo deep learning," *Image and Vision Computing*, vol. 87, pp. 47–56, 2019.
8. R. Laroca, E. Severo, L. A. Zanlorensi, L. S. Oliveira, G. R. Gonçalves, W. R. Schwartz, and D. Menotti, "A robust real-time automatic license plate recognition based on the yolo detector," in *2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–10, IEEE, 2018.
9. M. J. Shafiee, B. Chywl, F. Li, and A. Wong, "Fast yolo: A fast you only look once system for real-time embedded object detection in video," *arXiv preprint arXiv:1709.05943*, 2017.
10. W. Liu, C. Zhang, G. Lin, and F. Liu, "Crnet: Cross-reference networks for few-shot segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4165–4173, 2020.
11. F. Xie, M. Zhang, J. Zhao, J. Yang, Y. Liu, and X. Yuan, "A robust license plate detection and character recognition algorithm based on a combined feature extraction model and bpnn," *Journal of Advanced Transportation*, vol. 2018, 2018.
12. A. Nosseir and R. Roshdy, "Extraction of egyptian license plate numbers and characters using surf and cross correlation," in *Proceedings of the 7th International Conference on Software and Information Engineering*, pp. 48–55, 2018.
13. H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.
14. H. Sun, M. Fu, A. Abdussalam, Z. Huang, S. Sun, and W. Wang, "License plate detection and recognition based on the yolo detector and crnn-12," in *International Conference On Signal And Information Processing, Networking And Computers*, pp. 66–74, Springer, 2018.
15. R. Dey and F. M. Salem, "Gate-variants of gated recurrent unit (gru) neural networks," in *2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS)*, pp. 1597–1600, IEEE, 2017.
16. H. Li, P. Wang, M. You, and C. Shen, "Reading car license plates using deep neural networks," *Image and Vision Computing*, vol. 72, pp. 14–23, 2018.
17. M. Dong, D. He, C. Luo, D. Liu, and W. Zeng, "A cnn-based approach for automatic license plate recognition in the wild.," in *BMVC*, 2017.
18. Y. Alborzi, T. S. Mehraban, J. Khoramdel, and A. N. Ardekany, "Robust real time lightweight automatic license plate recognition system for iranian license plates," in *2019 7th International Conference on Robotics and Mechatronics (ICRoM)*, pp. 352–356, IEEE, 2019.
19. A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

20. L. Xie, T. Ahmad, L. Jin, Y. Liu, and S. Zhang, "A new cnn-based method for multi-directional car license plate detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 2, pp. 507–517, 2018.
21. J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.
22. J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7263–7271, 2017.
23. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.
24. G. A. Mills-Tettey, A. Stentz, and M. B. Dias, "The dynamic hungarian algorithm for the assignment problem with changing costs," *Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-07-27*, 2007.
25. C. Montella, "The kalman filter and related algorithms," *Adres: https://www. researchgate. net/. Alındığı tarih*, vol. 30, p. 2020, 2011.