Tanuj Kumar Kusmi, 2026, 14:3
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

Automatic Number Plate Detection Using Yolov8

Tanuj Kumar Kusmi[1], Manish Kumar[2], Vijay Kumar Gautam[3], Martand Mishra[4], Abishek Kumar Singh[5], Satyam Kumar[6]
[1]Distributed Systems Research Scientist, [2]Enterprise Platform Resilience Analyst,

Abstract- The traffic within cities has become worse. So we should implement automatic number plate recognition (ANPR). It can be used for traffic management, toll collection, police help and smarter. It has been proved that, some traditional image processing techniques and some of the deep learning approaches have not shown success due to their limitations like lighting condition, blocked license plates, and skewed license plates. This project proposes a system that utilizes YOLOv8. This system uses YOLOv8 and character recognition to detect license plates in the images and videos. So, it acts as an "eye" for identifying license plates. Automatic Number Plate Detection system, has demonstrated success in locating license plates, and it also performs efficiently under various lighting conditions including night, even under blocked plates. The combination of YOLOv8 and character recognition techniques offers a real-time solution for detection and identification of license plates in videos and images for effective traffic management and police assistance for instance tracing vehicles through[27]. The system has been designed to work for global number plates and can prove to be an effective solution for license plate detection with its fast speed technology. Good for Parking systems, toll collection and traffic management. Managing city traffic is a critical issue today, an effective Automatic Number Plate Detection system like the one using YOLOv8 and number plate could provide an efficient solution for this issue using number plates. Parking systems and toll collection can really benefit from this. Managing the traffic in a city is a problem nowadays. A good Automatic Number Plate Detection system that uses YOLOv8 and number plates can help solve this problem. work well with. To make the model work better some techniques have been used. These include flipping the image making it bigger or smaller and changing the brightness. This helps the model to work in different situations. The system can detect number plates fast and it is very accurate. It can detect number plates with an accuracy of 96.4%. It can process over 60 frames per second on a standard computer. This is very good, for Automatic Number Plate Detection systems that use number plates. The YOLOv8-based ANPR system does a job on benchmark datasets and real-world test scenarios. It is better than systems that use CNN or earlier YOLO versions when it comes to how fast it detects things how accurate it is and how well it works in different situations. The YOLOv8-based ANPR system can handle license plates that're at an angle partly covered or different sizes and it works well in various lighting conditions[16].

Keywords: Automatic Number Plate Recognition (ANPR), YOLOv8, License Plate Detection, Optical Character Recognition (OCR), Computer Vision, Deep Learning, Real-Time Detection, Traffic Management.

I. INTRODUCTION

There are more and more cars getting introduced on the road. This has resulted in major traffic management, border security, city planning issues. Automated Number Plate Detection systems also referred as Automated License Plate Recognition systems are one of the remedies for such problems[3][18].

The systems are capable of automatically detecting number plates from videos and images[4][20]. Some of the applications for automatic number plate detection systems are:

1). Toll collections
2). Parking management
3). Vehicle tracking
4). Identifying stolen cars
5). Speed enforcement / traffic surveillance

These automated number plate detection systems are very helpful to traffic management authorities. It can easily and quickly detect the number plates and help the police, government, authorities[21]. Automated number plate detection systems are used in management of parking areas, parking and toll collections. It also helps in vehicle tracking and identify stolen cars. Automated number plate detection systems are also very useful in effective speed enforcement and traffic surveillance.

Automated license plate recognition systems are also being introduced in the city. It helps the city planning authority in managing the traffic[6][7][18]. The automated license plate recognition systems are used for border security for quickly identifying vehicles passing the borders.

The number of automated number plate detection systems is increasing day by day, due to its ability to identify the number plate data at a much faster speed and also helping the authorities in more efficiently monitoring the vehicles. This is system is emerging as the key element in development of cities and smart cities are being developed on these automated number plate detection systems.

Prior to advanced detection of number plates via machine learning and deep learning models, traditional image processing methods such as template matching and edge detection worked "reasonably well"; however, suffered from certain limitations. Edge detection was susceptible to different conditions (changing light levels, different sizes of plates, weather) that would result in correctly identifying the plate information.

Machine learning models, specifically deep learning models utilizing CNN, have substantially improved the ability to accurately detect number plates through much better accuracy than traditional methods. On the other hand, first generation deep learning models (i.e. RCNN, Fast RCNN, SSD) experienced problems such as:

* slow processing time relative to speed of detection and
* poor accuracy during real-time detection operations.

The YOLO family of object detection models represented a significant wait for innovation in that the models significantly increased the speed in which objects were detected without compromising accuracy.

Since the original YOLO was launched, newer YOLO models have been developed with additional functionality and improved features.

* YOLOv8, developed by Ultralytics is one such model developed to deliver the ultimate performance in object detection systems.
* It utilizes a separate detection head, improved backbone architecture, and enhanced training methodology than past YOLO models.

This system has been designed with the intent of preparing the end-user for real world applications that have very high priorities on accuracy, speed and reliability.

By combining a YOLOv8 trained on a license plate database with Optical Character Recognition (OCR) for performance of character extraction, this research aims to provide an end-to-end automation number plate detection system directly related to intelligent transportation systems (ITS).

Number plate detection is critical for the operation of transportation systems.
Using YOLOv8 is one way that technology might accomplish this performance in environmental conditions and evaluates its suitability, for edge deployment.

The remainder of this paper is structured as follows:

1. The problem statement is presented in the section, Followed by the objectives of the study.
2. A comprehensive literature review surveys relevant works.
3. The proposed methodology and system architecture are then described in detail.
4. The analysis and results sections present findings,
5. The paper concludes with a discussion of contributions, limitations and future directions.

II. PROBLEM STATEMENT

Traditional number plate detection methods used image processing techniques like edge detection, morphological operations and template matching[21]. These methods worked okay in controlled environments. They had limitations in real-world situations with changing lighting, different plate sizes and fonts partial occlusions, weather variations and high-speed vehicle movement. The introduction of machine learning and deep learning techniques Convolutional Neural Networks (CNNs) improved detection accuracy significantly[9][22]. However early deep learning models like RCNN, Fast RCNN and SSD had slower inference times or lower accuracy in real-time streaming environments[6][7][16].

The YOLO (You Only Look Once) family of object detectors changed the game by making object detection a regression problem increasing inference speed while keeping accuracy competitive. Since the original YOLO was introduced new versions have been released, each improving on the last. YOLOv8, developed by Ultralytics is the most capable version. It uses an anchor- detection head, a new backbone architecture and improved training strategies[2] making it superior across various object detection benchmarks.

This research focuses on building and evaluating a number plate detection system using YOLOv8[12][20]. The proposed system targets real-world deployment scenarios where speed, accuracy and robustness equally critical. By training a YOLOv8 model on an diverse license plate dataset and integrating it with optical character recognition (OCR) for character extraction the research aims to deliver an end-to-end ANPD pipeline suitable for intelligent transportation systems. The study also looks at the models performance in environmental conditions and evaluates its suitability, for edge deployment.

The remainder of this paper is structured as follows:

1. The problem statement is presented in the section,
2. Followed by the objectives of the study.
3. A comprehensive literature review surveys relevant works.
4. The proposed methodology and system architecture are then described in detail.
5. The analysis and results sections present findings,
6. The paper concludes with a discussion of contributions, limitations and future directions.

The thing, with number plate detection is that it can make other problems worse when it is connected to recognizing the characters on the plate. If the system that finds the plate does not do a job it can hurt the quality of the picture of the plate that is used to recognize the characters. This can make it harder to get the characters. There is a problem because we do not have a good system that can find the number plate and then recognize the characters on it. We need a system that can find number plates quickly and correctly[1][10]. It has to work in many different situations. The number plate detection system has to be able to handle different kinds of plates and it has to be able to work with the character recognition system so that we can automatically identify the plates.

III. OBJECTIVES OF THE STUDY

The main goal of this study is to create an accurate Automatic Number Plate Detection system using the YOLOv8 deep learning framework[2]. This system should be able to detect license plates in time from videos and pictures taken in different environments. The system should have an accuracy of over 95% and be fast enough for use in traffic surveillance. The YOLOv8 framework will be used to develop the system. Another goal is to collect and prepare a dataset of license plate images from different places, vehicle types and environments. This dataset will be carefully. Enhanced using techniques like mosaic augmentation, random scaling, flipping and brightness adjustment. This will help the model learn to detect license plates and prevent it from becoming too specialized to a limited set of training data.

The study will also compare the performance of YOLOv8 with its versions and other object detection frameworks for detecting number plates. The comparison will look at metrics such as detection accuracy, speed, model size and computational requirements[6][10][11]. This will help justify why YOLOv8 was chosen as the detection backbone for this system.

The YOLOv8 framework is expected to perform in these comparisons. The results will provide evidence for the selection of YOLOv8. The study aims to show that YOLOv8 is suitable, for number plate detection.

We are proposing a solution, which comprises the YOLOv8 based plate detector and an OCR component to develop a complete number plate recognition system[12][20]. The objective of the developed number plate recognition system is to recognize all the numbers and characters of a number plate completely and accurately. This system must be robust and capable of correctly recognizing the number plate with above 93% accuracy under diverse circumstances. Additionally, the objective of this study is to evaluate system's robustness on computationally restricted embedded platforms such as NVIDIA Jetson, to investigate real-world applicability on computationally weaker surveillance platforms[2]. The YOLOv8 based plate detector needs to be robust and operate reliably on occluded, skewed or conditions.

IV. LITERATURE REVIEW

We propose a solution that includes a YOLOv8-based plate detector and an OCR component to create a number plate recognition system. The goal of this system is to recognize all numbers and characters on a number plate.

This system must work well. Recognize number plates with over 93% accuracy in different situations.

We also want to test the systems performance on devices with computing power like NVIDIA Jetson to see if it can work in real-world surveillance situations. The YOLOv8-based plate detector must be reliable. Work well with occluded, skewed or difficult conditions. Traditional methods used operators, filtering and analysis to find plate regions[21]. These methods worked well in controlled environments. Not in real-world situations with varying conditions.

The use of machine learning led to the development of Support Vector Machines (SVMs) and Boosting-based classifiers like Viola-Jones for Automatic Number Plate Recognition (ANPR) systems[21]. These methods improved detection by learning features from training data. Were limited by the quality of these features. Some methods used a sliding window approach with Histogram of Oriented Gradients (HOG) features, which worked well for plate images but failed with extreme variations in scale and angle. The introduction of learning, particularly convolutional neural networks improved object detection, including license plate localization.

Region-based CNN (RCNN) and its successors, RCNN and Faster RCNN provided substantial accuracy improvements. Several studies used RCNN for license plate detection achieving high accuracy; however the multi-stage architecture resulted in slow inference times.

Single-stage detectors, including SSD and RetinaNet offered inference but at some cost to detection accuracy, for small and occluded objects[7][17]. The YOLOv8-based plate detector and OCR component will help develop an accurate number plate recognition system. The systems robustness will be evaluated on restricted embedded platforms. The YOLOv8-based plate detector will operate reliably on occluded, skewed or difficult conditions.

YOLO detectors series work well on license plate detection in real-time. Yolov3 and Yolov4 were demonstrated with high accuracy and speed to detect license plates, making them applicable to traffic monitoring. Researchers added attention mechanisms, deformable convolutions and feature pyramid networks to YOLO to enable it to detect plates from long-distances. YOLOv5 achieved an even higher performance due to some architecture optimizations and superior data augmentation strategies. Many studies proved that YOLOv5 performs well in license plate detection in day and night conditions. Some techniques were employed for character recognition on the detected plates. Some researchers first segmented each character, then recognizing it with template matching or SVM classifier. Finally learning-based OCR techniques became state-of-the art; such CRNN utilizes CNN for feature extraction, LSTM for sequence modeling and CTC for decoding, and performs better on license plate character recognition when plates have diverse lengths. Ultralytics released YOLOv8 in 2023, which is the latest of YOLO detectors, it performs object detection without anchors; detection heads are used, C2f module is a novel backbone, it features task alignment strategies that lead to a significant increase in object detection performance, and it has been proved by some studies to outperform other detectors, however its performance and evaluation on real world license plate detection has been underexplored. Our study aims at filling this gap by evaluating YOLOv8 in number plate detection[2].

V. PROPOSED METHODOLOGY

The steps involved in auto number plate detection using YOLOv8 are discussed below in sequence to ensure that the system can effectively identify the number plate and be universally applicable. The system is comprised of five parts: Data acquisition and pre-processing preparing the data, setting up the YOLOv8 model and training, reading the characters using OCR, and testing the overall system[5].

The first step is to obtain a sufficient dataset of number plates. These number plates come from many datasets like CCPD and Open ALPR and from a set of custom images captured using cameras fixed on the roadside. The number plates we obtained came from many different regions around the world including Europe, India and America to ensure that the system performs in each of these conditions. These number plates are all captured in varied conditions as in day time, night time, varied weathers and viewing angles, the condition of being soiled or damaged[11].

For preparing the data we resize every picture to a uniform size of 640x640 which is the default input size for YOLOv8. All the pixel values are then normalized so that the system is not confused. Each picture is then annotated to define the location of the number plate on the picture in addition to information like the width and height of the plate[4]. During training some images are made more interesting to help the system learn to detect the number plates. For example, four images may be joined together, or an image may be horizontally flipped or the colour properties of the image altered to make it appear to have been captured in varying lighting conditions.strategies applied during training include mosaic augmentation combining four images into one, random horizontal flipping, random scaling and cropping, HSV color space augmentation to simulate lighting variation, and copy-paste augmentation for small object enhancement.

The YOLOv8 model we chose in this study is YOLOv8m version. It presents the compromise between precision and speed.

We used a pre-trained YOLOv8m model, trained on COCO dataset, then fined on our license plate dataset. The training was performed using Stochastic Gradient Descent optimizer, with cosine learning rate schedule and weight decay. We used precision training for accelerate training time.

The model trained over 150 epochs. The training process stopped if the accuracy for the validation dataset does not be improved. After detected plates, we used two-stage OCR pipeline for cropped plate image. The first stage helps for cleaning and un-skewing the plate image, the second stage reads the characters on plate using a CRNN-based OCR model. Our pipeline runs one frame in less than 30ms on an NVIDIA RTX 3060. Which is efficient enough for the traffic surveillance applications.

We evaluated the performance by object detection metric[22]. Those metrics include precision, recall, F1-score, MAP. The performance of character recognition was based on the number of plates whose all characters were recognized. We also performed ablation study to determine the effectiveness of each module of the system. Comparing our system to YOLOv5, YOLOv7 and

Faster RCNN. The models are compared under some settings.

VI. SYSTEM ARCHITECTURE

Proposed Automatic Number Plate Detection System is designed in a layered modular architecture. The system is designed with 6 layers or module. Those are data ingestion layer, preprocessing and augmentation layer, YOLOv8 detection engine, post processing and localization layer, OCR and character recognition module and the output and integration layer. The modular approach ensures that all these components are reusable and independent of other components[19]. We can individually update/modify one layer of this pipeline without affecting the other components. The data ingestion layer is the source to the system. This layer can intake information from any live camera streams, stored video files, still images. A multi-threaded frame capture logic is used to efficiently capture frames from multi cameras at the same time.

All the captured streams are decoded and buffered into frame queue which decouples the capture latencies from processing latencies. The frame queue also handles possible variable frame rate in live camera streams, especially for IP cameras. This layer is responsible for the basic preprocessing on the input frames. Normalizing the resolution, conversion to colour space, and basic quality check are performed on the frames. The resolution of frame will be normalized into 640x640 which is the input layer of the YOLOv8 detector. Normalize the pixels in a range of [0, 1]. In this layer, also a small frame quality check is included. We filter the frames which are seriously blurry or corrupt. This filter can minimize the fake detected plates. The same layer during training is used to conduct data augmentation, the full description can be seen in the methodology part[13].

The YOLOv8 detection engine will be the main part of the entire architecture. In this layer, the normalized frames are passed to the CSP-Darknet C2f backbone, extracting the feature of the frames with multiple scales. The feature map goes through the PAN neck, fusing the features of different scales together. Then the feature map is finally fed into the decoupled detection heads[22]. The detection heads predict the bounding boxes of the license plate and its class. Since the YOLOv8 is anchor-free, we do not need to care about anchor boxes any more. This layer involves the Non-Maximum Suppression algorithm to eliminate redundant bounding boxes of the same license plate and also uses some basic thresholds for filtering the low confidence and bounding box size smaller than a certain pixel. After filtering, the frame is cropped using the valid bounding box for OCR processing. However, we also apply perspective correction for the cropped license plate using homography transform.

The OCR and character recognition module gets pictures of plates that have been fixed to look right. It makes these pictures black and white to make them easier to read. The CRNN OCR model looks at the white picture and tries to figure out what characters are on the plate. It uses an algorithm to decode the characters and come up with the final plate number. The model also gives a score for how sure it's about each character it recognizes so if it is not very sure someone can check it by hand[23].

The output and integration layer puts the recognized plate numbers with where they were found on the picture how sure the model is and when it was found. It then sends this information to systems that need it like parking management systems, toll collection systems or traffic violation databases using a special kind of messaging system. The way the API is designed makes it easy to connect it to systems that are already being used in cities and, for transportation. There is also a part that saves records of what was found in a database so it can be looked at later for checking or analysis[25].

VII. ANALYSIS

The Automatic Number Plate Detection system seems to be a bargain then. We need to measure how effective this system really is so we conducted experiments by testing it in various locations. One of these was a control environment with varying light intensities, another one was a parking lot outdoors with varying distances and angle of vehicles and finally one was a highway with quickly passing vehicles under natural light intensities[27].

The YOLOv8m system performed brilliantly when it came to detecting the number plates. Average Precision (AP) scored 96.4% on the main test set which means it accurately identifies number plates in most of the images. We then examined how the model performed on varying criteria of overlap and got AP 78.3% which indicates that the model is good at identifying number plates which are not exactly centre in the picture. Precision was 97.1% and Recall was 95.6% which indicates that the system has low false positives and false negatives.

We also examined how the system performs in varying light conditions. It works quite effectively when it comes to capturing number plates at night, illuminated by the lights of the cars and the streetlights, and in case of infrared light. The sun reflection sometimes proves to be quite tricky and hence in conditions of direct sunlight when the number plate reflects the light of the sun, the system is just adequate[16].

We also investigated how the system works when the number plates were at an angle from the camera. The model performs brilliantly when the plate is turned at a maximum angle of 35 degrees, then the image is distorted and the performance reduces. The system also performs well when the number plate is at an upward and downward angle up to 25 degrees. All the observed angles are common for cameras placed on side of the roads. All in all the Automatic Number Plate Detection system works perfectly fine[10].

I have studied and analyse, how YOLOv8 works in comparison with YOLOv5, YOLOv7, and Faster RCNN. The same data set, evaluation settings were used for all of them. I found out that YOLOv8 wins in all these areas compared with YOLOv5, YOLOv7, and Faster RCNN. YOLOv8m has a 3.2% gain over YOLOv5m with respect to detection, and 1.8% over YOLOv7. These gains are achieved with relatively high speed. Faster RCNN has decent performance regarding detection but has really low speed. It has 8 frames per second only. YOLOv8m performs 63 frames per second on the PC. That makes YOLOv8m better for applications in real-time. I've also checked the points where YOLOv8 struggles. It struggles with blocked plates and just a small portion of the plate visible. It also struggles with low resolution due to large distance of the camera. Dirty or damaged plates can also cause trouble for YOLOv8. These are the problems which should be overcome by YOLOv8. To make YOLOv8 improve we should collect some images with this kind of plates and make them more clear with some techniques[23].

VIII. RESULTS

Overall, the performance of the Automatic Number Plate Detection system with YOLOv8 is good. The system can detect license plates in image and video with in time with accurate predictions. YOLOv8 system performance on this is great[30].

The system has been evaluated on the 4,200 conditions images. The accuracy of the YOLOv8m system is impressive, it achieved an Average Precision of 96.4%, Precision of 97.1%, Recall of 95.6% and an F1-Score of 96.3% which significantly outperforms the YOLOv5, YOLOv7 and Faster RCNN system in term of detection accuracy. It shows that YOLOv8 can be a suitable candidate in Automatic number plate detection.

Additionally, the system performance can be quite fast. It achieves 63.4 FPS on NVIDIA RTX 3060 GPU and 28.7 FPS on NVIDIA Jetson Orin edge devices. The YOLOv8 based Automatic Number Plate Detection system works in time.

Table 1: Comparative Performance of Detection Models

Model	**mAP@0.5 (%)**	**Precision (%)**	**Recall (%)**	**FPS**
Faster RCNN	95.1	95.8	94.2	8.3
YOLOv5m	93.2	93.9	92.4	47.6
YOLOv7	94.6	95.2	93.7	54.2

Model	mAP@0.5 (%)	Precision (%)	Recall (%)	FPS
YOLOv8m (Proposed)	96.4	97.1	95.6	63.4

The license plate recognition system did a job and was able to read the license plates about 94.3 percent of the time. It was even able to recognize the characters on the license plates with an accuracy of 98.7 percent of the time. The license plate recognition system performed better when it saw plates with letters and numbers getting it right about 96.1 percent of the time. However, the license plate recognition system did not do well with plates that had two languages getting it right about 91.4 percent of the time. The license plate recognition system also worked well at night detecting plates 92.8 percent of the time which means the license plate recognition system can be used to watch things all day and all night. The license plate recognition system worked well on a computer called the NVIDIA Jetson Orin. The license plate recognition system was able to look at 28.7 frames per second which is better than the 25 frames per second that makes video look smooth. The license plate recognition system did not use a lot of power using about 12.3 watts and the license plate recognition system can even be used with panels to watch the road without being plugged into the main power grid[29].

The license plate recognition system has parts that help it do its job and if some of these parts were missing the license plate recognition system would not work as well. For example if the license plate recognition system was not trained using pictures that're a little distorted it would not be as good at recognizing license plates. Also if the license plate recognition system cannot adjust the picture correctly it will not work well.. If the license plate recognition system has seen the picture before it will do a good job. It is important to understand how the license plate recognition system works so we can make it better, in the future, which shows that the license plate recognition system is an useful tool[25].

IX. CONCLUSION

This research is about a Number Plate Detection system that uses the YOLOv8 deep learning framework. It shows that we can get good detection accuracy and fast performance at the same time, which is great for traffic surveillance. The system works with different lighting, weather and license plate formats and it can even handle cars moving really fast. It gets an Average Precision of 96.4% and an end-to-end recognition accuracy of 94.3%, which is very good.

The experiments show that YOLOv8 is better than systems like YOLOv5, YOLOv7 and Faster RCNN. YOLOv8 is good because it does not use anchors for detection it has a C2f backbone. It has a task-aligned assignment strategy. This makes it very good for detecting number plates. We compared YOLOv8 to systems and it did better in terms of accuracy and speed. This means it is a choice for next-generation ANPR systems[21].

We also tried combining YOLOv8 with techniques like perspective correction and CRNN-based OCR. This worked well. Showed that using a modular pipeline design can give us high recognition performance. The perspective correction stage is especially helpful when the license plate is not facing the camera directly. This happens a lot in traffic surveillance. We found that it improved recognition accuracy by 4.2%. We tested the system on NVIDIA Jetson Orin hardware. It worked well even with limited resources. This is important because it means we can use the system in transportation infrastructure without needing a big computer server. This can help reduce costs and make the system more practical, for large-scale use[24].

The new system is really good at what it does. It has some problems though. For example it does not work well with number plates that're hard to see or have many different scripts on them. We can work on making the system better in these areas. We need to get data, from many different places. We should also try using ways of detecting number plates like using transformer architectures. This will help make the system better at recognizing number plates automatically[26].

REFERENCES

1. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," Proceedings of CVPR, pp. 779-788, 2016.
2. G. Jocher et al., "Ultralytics YOLOv8," Ultralytics GitHub Repository, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics
3. A. 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," Proceedings of IJCNN, pp. 1-10, 2018.
4. X. Li, D. Lv, and J. Yi, "An Efficient License Plate Recognition System Based on YOLO and LPRNet," IEEE Access, vol. 10, pp. 75284-75295, 2022.
5. Z. Liu, W. Liu, and C. Qiao, "A New Approach for License Plate Detection Based on Improved YOLOv5," Sensors, vol. 22, no. 3, pp. 1212, 2022.
6. S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137-1149, 2017.
7. W. Liu et al., "SSD: Single Shot MultiBox Detector," Proceedings of ECCV, pp. 21-37, 2016.
8. Z. Xu, W. Yang, A. Meng, N. Lu, H. Huang, C. Ying, and L. Huang, "Towards End-to-End License Plate Detection and Recognition: A Large Dataset and Baseline," Proceedings of ECCV, pp. 261-277, 2018.
9. Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," Nature, vol. 521, pp. 436-444, 2015.
10. A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," arXiv preprint arXiv:2004.10934, 2020.
11. C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors," Proceedings of CVPR, pp. 7464-7475, 2023.
12. 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 PAMI, vol. 39, no. 11, pp. 2298-2304, 2017.
13. K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," Proceedings of CVPR, pp. 770-778, 2016.
14. T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature Pyramid Networks for Object Detection," Proceedings of CVPR, pp. 2117-2125, 2017.
15. A. Vaswani et al., "Attention Is All You Need," Advances in Neural Information Processing Systems, vol. 30, pp. 5998-6008, 2017.
16. R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," Proceedings of CVPR, pp. 580-587, 2014.
17. T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal Loss for Dense Object Detection," Proceedings of ICCV, pp. 2980-2988, 2017.
18. A. Laroca, L. A. Zanlorensi, G. R. Gonçalves, E. Todt, W. R. Schwartz, and D. Menotti, "An Efficient and Layout-Independent Automatic License Plate Recognition System Based on the YOLO Detector," IET Intelligent Transport Systems, vol. 15, no. 4, pp. 483-503, 2021.
19. J. Guo, Z. Liu, and L. Liu, "Key Information Extraction from License Plate Using Deep Learning," Proceedings of ICME, pp. 1-6, 2020.
20. C. Henry, S. Y. Huang, and Y. Lim, "End-to-End Learning for License Plate Detection and Recognition," Proceedings of ICPR, pp. 4102-4107, 2020.
21. N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," Proceedings of CVPR, vol. 1, pp. 886-893, 2005.
22. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," Advances in Neural Information Processing Systems, vol. 25, pp. 1097-1105, 2012.
23. M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes (VOC) Challenge," International Journal of Computer Vision, vol. 88, pp. 303-338, 2010.
24. T.-Y. Lin et al., "Microsoft COCO: Common Objects in Context," Proceedings of ECCV, pp. 740-755, 2014.

25. G. Huang, Z. Liu, L. Van der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," Proceedings of CVPR, pp. 4700-4708, 2017.
26. S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp. 1735-1780, 1997.
27. K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," Proceedings of ICLR, 2015.
28. C. Szegedy et al., "Going Deeper with Convolutions," Proceedings of CVPR, pp. 1-9, 2015.
29. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," Proceedings of CVPR, pp. 4510-4520, 2018.
30. D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," Proceedings of ICLR, 2015.