Personal and Ubiquitous Computing
https://doi.org/10.1007/s00779-020-01514-z

ORIGINAL ARTICLE

Research of automatic recognition of car license plates based on deep learning for convergence traffic control system

Hyochang Ahn[1] · Han-Jin Cho[1]

Received: 29 May 2020 / Accepted: 29 December 2020

Abstract
The technology that can recognize the license plates of vehicles in real time and manage them automatically is a key element of building an intelligent transportation system. License plate recognition is the most important technique in vehicle image processing used to identify a vehicle. Object recognition using a camera is greatly influenced by environmental factors in which the camera is installed. When the vehicle image is acquired, the image is distorted due to the tilting of the license plate, reflection of light, lighting effects, rainy weather, and nighttime, so that it is difficult to accurately recognize the license plate. In addition, when the geometric distortion of the license plate image or the degradation of the image quality is intensified, it may be more difficult to automatically recognize the license plate image. Therefore, in this paper, we propose a deep learning–based vehicles' license plate recognition method to detect license plate and recognize characters accurately in complex and diverse environments. As a deep learning model, the YOLO model can be used to detect robust license plates in a variety of environments and to recognize characters quickly and accurately. It can also be seen that the license plate accurately recognizes the license plate with geometric distortion.

Keywords Deep learning · License plate · Recognition · Intelligent transportation system

1 Introduction

The demand for vehicles is increasing globally due to economic growth and income increase, as well as the increasing demand for private cars. As vehicles increase, there are various problems to be solved by automobile-related policies as well as automobile-related industries [1–3]. Problems are solved or supplemented by the development of technology, but there are still many problems to be solved. Many efforts are being made to manage poor traffic management systems with limited resources.

Autonomous driving cars are also one of the biggest social issues. Many autonomous vehicles have many technologies such as lane recognition, vehicle distance recognition, and object recognition [4]. Most of these technologies are based on cameras. License plate recognition (LPR) is the most widely used technology related to automobiles using such a camera. The technology that can recognize the license plates of vehicles in real time and manage them automatically is a key element of building an intelligent transportation system. License plate recognition is the most important technique in vehicle image processing used to identify a vehicle. Vehicle license plate recognition technology is currently used in various fields. In particular, it is used for a lot of purposes such as parking lot, intelligent transport system (ITS) or smart highway, toll collection, violating vehicle detection [5, 6]. For this purpose, the technique of accurately recognizing the position of the license plate is particularly important.

The license plate recognition algorithm is generally divided into license plate area detection and number recognition process [7–9]. License plate area detection is to recognize a vehicle in an image and to detect a region of interest that is determined as a license plate area in the vehicle. Currently, most license plate recognition mainly adopts a single license plate recognition method in a fixed position. Object recognition using a camera is greatly influenced by environmental factors in which the camera is installed. When the vehicle image is acquired, the image is distorted due to the tilting of the license plate, reflection of light, lighting effects, rainy

✉ Han-Jin Cho
hanjincho@kdu.ac.kr

Hyochang Ahn
youcu92@kdu.ac.kr

[1] Department of Energy IT Engineering, Far East University, Eumseong, Chungbuk, Republic of Korea

Published online: 27 February 2021

weather, and night time, so that it is difficult to accurately recognize the license plate [10–12]. In particular, when the geometric distortion of the license plate image or the degradation of the image quality is intensified, it may be more difficult to automatically recognize the license plate image. If these results are output in situations where reliability is very important, they will be fatal. To prevent this result, a technology that can minimize environmental factors and maintain almost constant image quality is required. Number recognition is to recognize numbers and characters in the detected license plate area. Car number recognition is more affected by the surrounding environment than other character recognition technologies [13–16]. Therefore, researches of license plate recognition technology should be considered to be strong against noise and deformation. However, due to the nature of the license plate, the characters used are limited, and because the character is constant, the complexity is lower than other character recognition. In addition, OCR-net uses a weak modification of the YOLO network to better recognize license plates in various countries. OCR-net uses a model that learns the character fonts and license plates of license plates from various countries. In this paper, we also recognize car letters based on OCR-net.

Therefore, in this paper, we propose a deep learning–based car license plate character recognition method to detect the car's license plate and recognize numbers accurately in complex and diverse environments. In particular, we propose a method of accurately recognizing license plates with geometric distortion with a deep learning method. As a deep learning model, the YOLO model can be used to detect robust license plates in a variety of environments and to recognize numbers quickly and accurately.

The structure of this paper is discussed in Section 2 for the method related to license plate recognition. In Section 3, license plate recognition using the deep learning method is described. In Section 4, we compare and analyze the experimental results of recognizing license plates with deep learning. Finally, we conclude in Section 5.

2 Related work

2.1 License plate detection

Automotive license plate area detection is a technique for finding license plate areas before recognizing license plate characters in an image. The reason for finding the license plate area first is raw data for recognizing and character recognition of the license plate area. In some cases, the image is preprocessed to increase the detection performance of the license plate area. Representative methods for detecting a license plate area include an edge method, an outline method, and a neural network method [8, 9, 17].

2.1.1 License plate area detection using edges

The edge-based license plate detection method converts a color vehicle image into a gray image, passes a blur filter, and applies the Sobel filter horizontally to reduce the phenomenon of excessive edges. Then, binarization is performed to project in the horizontal direction [9, 18]. The number plate has numbers and characters, so that the projection value rises specifically to the location of the license plate string, finds the horizontal value of the license plate position, applies the Sobel filter vertically in the same way, and detects the license plate area by Hough transform.

In the edge-based license plate area detection method, when the steps of removing unwanted edges from the processed image are combined, the license plate extraction rate shows a relatively good performance compared to other methods. However, when the image background is complicated, it is difficult to find the position of the license plate pattern among the projected results, and the front radiator grille of the car may be recognized as the license plate. There is a disadvantage in that the license plate cannot be extracted if it exceeds a certain detection range. Figure 1 shows a license plate area extraction diagram using edges.

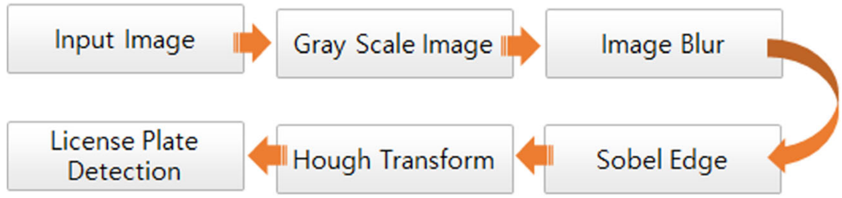
2.1.2 License plate area detection using contour

In order to find the outline of the license plate, the color vehicle image is transformed using gray image conversion, Gaussian blur filter, vertical Sobel filter, binarization, and morphology closing operator. The license plate is found by finding a contour corresponding to the size and aspect ratio of the license plate in the converted image [19, 20]. This method detects license plates by reducing the light and dark background parts of the license plate relatively. If the character part of the license plate is clearly different from the background part, it is easy to extract it separately. However, there is a disadvantage in that the exact license plate cannot be extracted in the case where the background is complicated, such as an image in which characters and backgrounds are difficult to distinguish from light or an image in which the license plate is tilted. Figure 2 shows a license plate area extraction diagram using contour.

2.1.3 License plate area detection using neural network

In vehicle license plate recognition, the neural network was mainly used to find the license plate and then to recognize the characters and numbers on the license plate. In some studies, however, neural networks have also been used to detect license plate features [7, 21]. The filter analyzes the color and texture characteristics of a particular window of the input image and classifies it into two cases: part of the license plate of the pixel in the center of the window or background. The

Fig. 1 License plate area extraction diagram using edges



license plate is finally extracted through the post-processing process from the image passing through the filter. The method of using a filter to analyze the color or texture of an image is widely known, and when used as a filter, a neural network learning algorithm can be used instead of a complicated filter parameter estimation process. Therefore, a good effect can be expected from the advantage that the noise is less affected by the characteristics of the neural network. These methods allow for relatively accurate license plate extraction. However, since the target image is a color image, there is a large amount of information to be processed, and the time required in consideration of the process of adding the learning time and recognition of the license plate character due to the use of neural networks.

2.2 Character recognition

License plate character recognition is a technique of recognizing a character area extracted from a detected license plate area. The license plate character in the image is recognized using the font characteristics and standard characteristics. In general, character recognition techniques are frequently used in license plate character recognition techniques. License plate character recognition is not very different from general character recognition, but considering the characteristics of license plate character, performance can be better than general character recognition. Representative methods for character recognition include a method using template matching and a method using pattern recognition [6, 7, 22, 23].

2.2.1 Character recognition using template matching

There are various methods of character recognition technology, but the simplest of them is template matching. Template matching is a method of searching for a location matching a template image in a target image [6, 7]. The template image is placed on the reference image, and the pixel values of the portion of the reference image covered with the template image and the pixel values of the template image are compared by a specific mathematical operation. In this way, the template image is moved to compare the entire reference image. In this way, a coefficient of correlation map is obtained, and a value having the maximum similarity is found in the correlation coefficient map and selected as the detected character. Character recognition using template matching is very easy to implement, but it has the disadvantage of being very vulnerable to various variables such as contrast, noise, tilt, and rotation of the character.

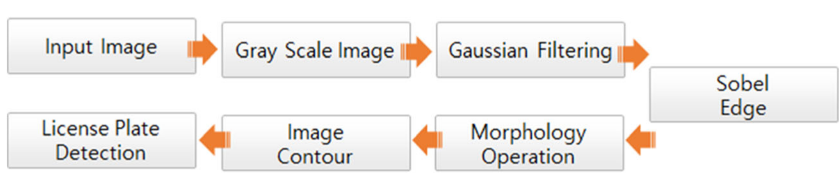
2.2.2 Character recognition using pattern recognition

The pattern recognition system can be largely divided into a part for marking a pattern and a part for analyzing a pattern. A given pattern must be written in the same way as the pattern prototype to identify a match [7, 22, 23]. The problem of pattern recognition can be solved in several ways. The first is a static approach and the second is a syntactic approach. Among them, identification in statistical pattern recognition is a method of mapping a pattern to a feature space and then learning an identifier that can determine a class of a newly input pattern. However, when the image pixel is used as a feature, the dimension of the pattern becomes high, and the curse of dimensionality occurs that information overlaps between features and processing time increases. Therefore, in the case of pattern recognition in order to improve the performance of the identifier when the feature is higher than the number of patterns, a process of reducing the pattern feature is required. Principal component analysis (PCA) and linear discriminant analysis (LDA) are widely known as a method of reducing the feature space to improve the identification performance [24, 25].

3 License plate recognition using deep learning

Various methods have been developed and commercialized in real life for vehicle license plate detection in good lighting and

Fig. 2 License plate area extraction diagram contours



pre-made image acquisition environment. However, detecting and recognizing a license plate in real-time image are affected by the surrounding environment. Geometric distortion of characters or numbers occurs depending on the distance between the surveillance camera and the license plate [13–16]. In addition, it is difficult to distinguish between a character and a background due to lighting or light reflection. Therefore, researches of license plate recognition technology should be considered to be strong against noise and deformation.

In this paper, we propose a deep learning–based car number recognition method to detect the license plate of a car and recognize the number accurately in complex and diverse environments. In particular, we propose a method of accurately recognizing license plates with geometric distortion with a deep learning method. The YOLO model can be used as a deep learning model to detect robust license plates and perform number recognition in various environments.

3.1 Deep learning with YOLO

YOLO is an object recognition API built on GoogLeNet. GoogLeNet has a CNN structure but a deeper structure [26]. R-CNN-based deep learning for object detection algorithms, such as Faster R-CNN, had a two-stage structure in which bounding boxes were first found and then classified for each bounding box [27, 28]. YOLO algorithm is the first one-shot architecture proposed to improve the speed of the object recognition algorithm. Since the process of one-shot architecture processes the image only once with one CNN, the area and object area suggestion is not necessary, so it is simpler and much faster than the existing R-CNN-based algorithm [29–31]. Figure 3 shows the image recognition process using YOLO.

YOLO has the advantage that high-speed processing is possible by applying one neural network directly to the whole image, unlike the conventional method of obtaining the object bounding box and recognizing objects along various positions and scales while scanning in the image [32–34]. YOLO first divides the input image into a grid to predict the bounding box within the grid then maximizes the confidence in the box and detects the probabilistic classification of the classes for that object. The result is the detection of the center position coordinates (x, y), height and width (w, h) of the object, and at the same time probabilities of confidence in the type of the object.

A final detection area is determined by constructing a plurality of predicted detection areas and images with a grid 7 × 7 through the CNN structure of the input image. Each grid consists of a label for the class and the coordinates of the object to predict the difference between the learning area coordinates and the predicted coordinates, and the probability value for the class when the object is detected in the grid.

YOLO model uses a loss function to find the optimal weight parameter through training. This loss function consists of three kinds: localization loss, confidence loss, and classification loss.

The localization loss function can be expressed as the difference between the measured and expected coordinates of all cells in the image and the objects within the cell, the measured width and height of the bounding box, and the difference between the expected width and height. Equation (1) represents the localization loss function.

$$\lambda_{\mathrm{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{ij}^{\mathrm{obj}} \left[\left(x_i - \widehat{x}_i \right)^2 + \left(y_i - \widehat{y}_i \right)^2 \right] + \lambda_{\mathrm{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{ij}^{\mathrm{obj}} \left[\left(\sqrt{\omega_i} - \sqrt{\widehat{\omega}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\widehat{h}_i} \right)^2 \right] \tag{1}$$

The confidence loss function is a value calculated from the trust loss function to determine whether all grid cells and the bounding box in them are similar to real objects. Equation (2) represents the confidence loss function, and if the object is not found, the loss value is reduced to a value.

$$\lambda_{\mathrm{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{ij}^{\mathrm{noobj}} \left(C_i - \widehat{C}_i \right)^2 \tag{2}$$

The classification loss function can be expressed as the square of the difference between the probability of a class in each object and the conditional class probability when an object is detected. Equation (3) represents the classification loss function, which represents a value of 1 if the object is in a cell and a value of 0 if it is not.

$$\sum_{i=0}^{S^2} I_{ij}^{\mathrm{obj}} \sum_{c \in \mathrm{classes}} \left(p_i(c) - \widehat{p_i(c)} \right)^2 \tag{3}$$

Equation (4) represents the total loss function of YOLO and is designed to calculate the class of the object to be detected, the size of the location and bounding box, and the existence of the object per grid.

$$\lambda_{\mathrm{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{ij}^{\mathrm{obj}} \left[\left(x_i - \widehat{x}_i \right)^2 + \left(y_i - \widehat{y}_i \right)^2 \right] + \lambda_{\mathrm{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{ij}^{\mathrm{obj}} \left[\left(\sqrt{\omega_i} - \sqrt{\widehat{\omega}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\widehat{h}_i} \right)^2 \right] + \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{ij}^{\mathrm{obj}} \left(C_i - \widehat{C}_i \right)^2 + \lambda_{\mathrm{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{ij}^{\mathrm{noobj}} \left(C_i - \widehat{C}_i \right)^2 + \sum_{i=0}^{S^2} I_{ij}^{\mathrm{obj}} \sum_{c \in \mathrm{classes}} \left(p_i(c) - \widehat{p_i(c)} \right)^2 \tag{4}$$

The input image size of YOLO is 416 × 416, and if you use a different size image, it is resized to 416 × 416. The output of each image is $S \times S \times (B \times 5 + C)$ values. $S \times S$ means the

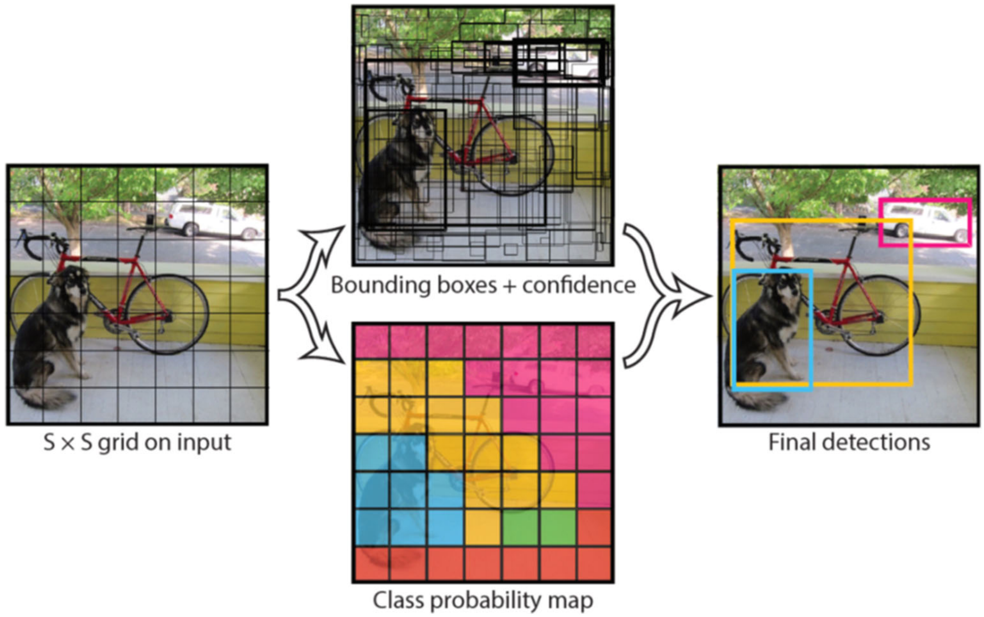


Fig. 3 Object recognition processing using YOLO

number of grids and $S = 7$ by default. B is the number of bounding boxes that YOLO predicts, $B = 2$ is applied, and C is the number of classes YOLO will detect.

The advantage of YOLO is that the detection problem is regarded as a regression analysis, which does not require a complicated computational process. Neural network structure of YOLO is shown in the figure, and filter size and number of stride parameters are written. In Fig. 4, the FC is a fully connected layer.

The convolutional neural network model applied to YOLO consists of convolutional and maximal pooling layers, replacing the fully connected layer with convolutional layers, while maintaining a mechanism for predicting the position and class of objects at once. In addition, the size of the input image for YOLO's learning was set to 416 × 416 in order to determine the final output size through the multiplication neural network divided by 32 by 13 × 13. The final output size was determined to be 13 × 13 so that the middle grid cell was centered. This is because, when the object is in the center, the prediction method with one grid cell is more efficient than the prediction with four grid cells with respect to the center. YOLO applied anchor boxes to predict more than 1000 bounding boxes, and predicted objectivity and class for each anchor box. This improves the performance of bounding box prediction for objects of various shapes.

3.2 Car license plate recognition with YOLO

In YOLO, it is necessary to improve the bounding box prediction performance by optimizing the anchor box according to the shape of the object in the given image data. The size of the anchor box was optimal using the K-means clustering

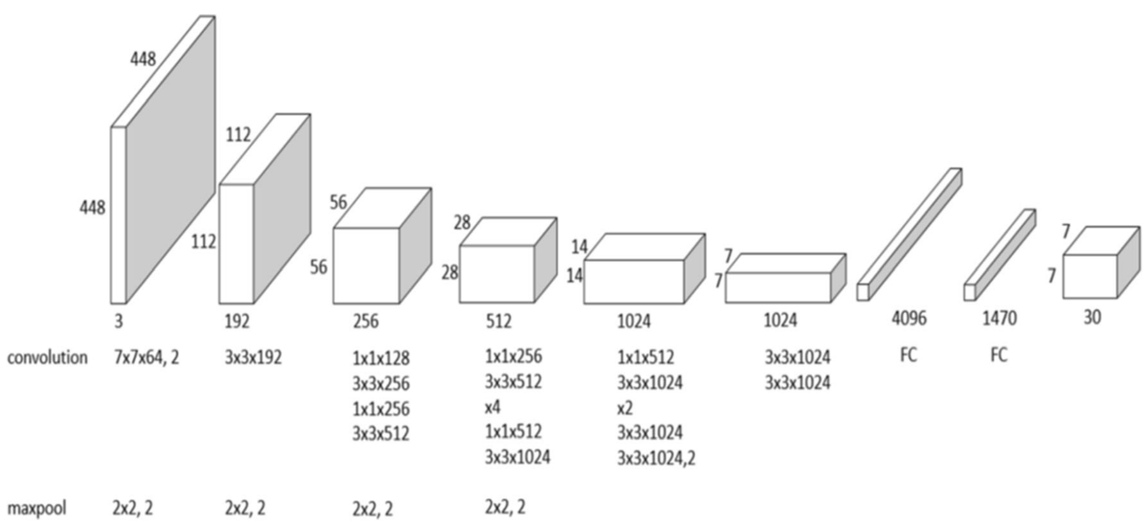


Fig. 4 Neural network structure of YOLO

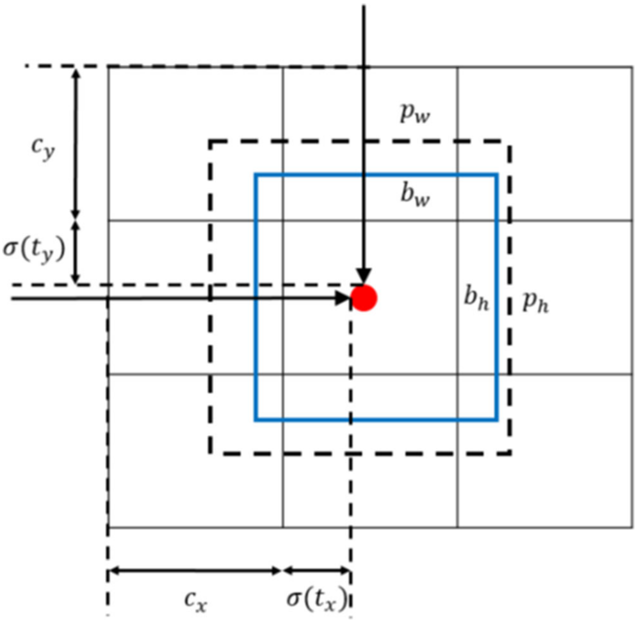


Fig. 5 Bounding box coordinate prediction with YOLO

technique. At this time, if standard *K*-means clustering is used as it is, a large anchor box generates more loss than a small anchor box. Therefore, by applying *K*-average clustering, clustering data having a close distance between data with respect to the horizontal and vertical length of the anchor box can obtain the size of the anchor box optimized for the data set.

When applying *K*-means clustering, determining *K*, which is the number of centroids, has the advantage of improving recognition performance. However, the number of anchor boxes that need to be predicted increases the amount of computation for object detection, which can slow down object detection.

YOLO predicts coordinates based on each grid cell. The coordinates of the bounding box can be predicted in the same way as in Fig. 5.

The range of the correct answer value is between 0 and 1 so that the center coordinate of the object exists inside each grid cell. The logistic activation function is used for the predicted value. Equation (5) represents a logistic activation function.

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{5}$$

3.3 Character recognition with YOLO

The general license plate character recognition step involves two processes. The first step is to segment each character and then classify the characters one by one. But going through two steps is time-consuming. In addition, if you segment the characters incorrectly in the first step, the entire process of recognizing the number plate character is invalidated. Recently, since one character is assumed to be a class of objects, the character recognition problem has turned into an object recognition problem. Therefore, the letters A–Z and numbers 1–9 are assumed to be all 35 classes. The number is assumed to be a letter where 0 is O. There is a research to make OCR-net using a weak modification of YOLO network. To better recognize license plates in various countries, OCR-net learned the character fonts and license plates of license plates in many countries. This paper also recognizes OCR-net based characters.

4 Experiment result

In this paper, we evaluate the performance using Python in Windows 10 environment. A learning model was created through YOLO learning of the data set and license plate detection and character recognition were performed on the images. License plate character recognition is a modified version of the YOLO model, which uses OCR-net to recognize characters. Figure 6 shows a diagram for detecting license plates and recognizing characters in an image. Figure 7 shows the license plate data set used in the experiment.

Confusion matrices were used to evaluate the performance of the experiment. The convergence matrix is a table that can visualize the performance of classification algorithms trained in supervised learning in problems such as statistical classification in the field of machine learning. Each row of the matrix represents an instance of the predicted class, and each column represents an instance of the actual class. The name Confusion comes from the fact that it is easy to see how confusing the two classes are to the system. Table 1 shows the fusion matrix used in the experiment.

When evaluating the object detection performance, it is evaluated through various indicators such as Precision, Recall, Average Precision, and mAP. Precision is the ratio of Predicted Condition Positives to condition positives. Equation 6 represents precision.

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \tag{6}$$

Fig. 6 License plate detection and character recognition diagram

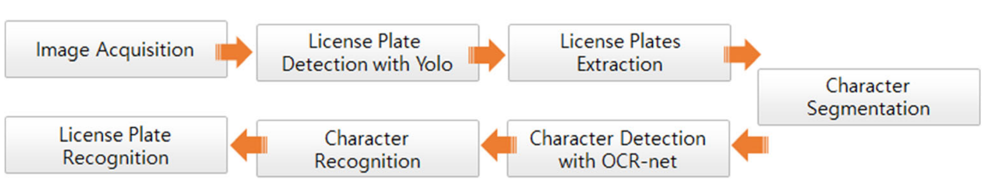


Fig. 7 Sample data for learning license plates

Recall is the ratio of predicted Predicted Condition Positive out of actual condition positive. Equation 7 represents recall.

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \quad (7)$$

If both the Precision and Recall values are good, it is a good model, but it is usually difficult to decide whether we will evaluate the model with Precision or only with Recall. So, we use Average Precision (AP) to compensate for the problems associated with Precision and Recall values. Each time the threshold is converted, you can get the value of the

Table 1 Confusion matrices

		True condition	
		Condition positive	Condition negative
Predicted condition	Predicted condition positive	True positive	False positive
	Predicted condition negative	False negative	True negative

Table 2 License plate recognition performance comparison

	Average precision
K-NN	89.82
Faster R-CNN	91.1
YOLO	95.7

Precision and Recall pairs, and you can get a Precision-Recall curve by plotting each point on the graph. Equation 8 shows Average Precision.

$$\text{Average Precision} = \sum_{n} (R_n - R_{n-1}) \times P_n \tag{8}$$

Experimental results show that the average precision of the license plate recognition model is 95.7%, which shows that the deep learning–based car number recognition using YOLO is improved compared to other methods. Table 2 shows the experimental results compared with the conventional method. Figure 8 also shows the license plate recognition results using YOLO.

5 Conclusions

A lot of researches have been done on the license plate recognition, and license plate recognition methods that have excellent performance under certain conditions such as parking lot

Fig. 8 License plate recognition results using YOLO

or road have been commercialized. The license plate recognition method mainly used up to now consists of finding a license plate area in a car image and recognizing numbers and letters in the license plate area. In order to solve these two stages, researches to improve the recognition speed and recognition rate have been actively conducted.

When the vehicle image is acquired, the image is distorted due to the tilting of the license plate, reflection of light, lighting effects, rainy weather, and night time, so that it is difficult to accurately recognize the license plate. In particular, when the geometric distortion of the license plate image or the degradation of the image quality is intensified, it may be more difficult to automatically recognize the license plate image.

Therefore, in this paper, we propose a deep learning–based car number recognition method to detect car's license plate and recognize numbers accurately in complex and diverse environments. As a deep learning model, the YOLO model can be used to detect robust license plates in a variety of environments and to recognize numbers quickly and accurately. In addition, OCR-net uses a weak modification of the YOLO network to better recognize license plates in various countries. OCR-net uses a model that learns the character fonts and license plates of license plates from various countries. In this paper, we also recognize car letters based on OCR-net. Experimental results show that the average precision of the license plate recognition model is 95.7%, which shows that the deep learning–based car number recognition using YOLO is improved compared to other methods.

In the future, we need to study the recognition of license plates at the same time for multiple vehicles and research on how to recognize the license plates in low-resolution images.

References

1. Du S, Ibrahim M, Shehata M, Badawy W (2013) Automatic license plate recognition (ALPR): a state-of-the-art review. IEEE Trans Circ Syst Video Technol 23(2):311–325
2. Robert K (2009) Video-based traffic monitoring at day and night vehicle features detection tracking, 2009 12th International IEEE Conference on Intelligent Transportation Systems. IEEE, pp. 1–6
3. Naito T, Tsukada T, Yamada K, Kozuka K, Yamamoto S (2000) Robust license plate recognition method for passing vehicles under outside environment. IEEE Trans Veh Technol 49(6):2309–2319
4. An KH, Lee SW, Han WY, Son JC (2013) Technology trends of self-driving vehicles. ETRI Electron Telecommun Trends 28(4): 35–44
5. Montazoli S, Jung C (2018) License Plate Detection and recognition in unconstrained scenarios, Proceedings of the European Conference on Computer Vision (ECCV). pp580–596, 2018
6. Anagnostopoulos C, Alexandropoulos T, Boutas S, Loumos V, Kayafas E (2005) A template-guided approach to vehicle surveillance and access control, IEEE Conference on Advanced Video and Signal Based Surveillance, pp.534–539
7. Zheng D, Zhao Y, Wang J (2005) An efficient method of license plate location. Pattern Recogn Lett 26(15):2431–2438
8. Lee ER, Kim PK, Kim HJ (1994) Automatic recognition of a car license plate using color image processing. Proc 1st Int Conf Image Process 2:301–305
9. Taktak R, Dufaut M, Husson R (1994) Road modeling and vehicle detection by using image processing, Proceedings of IEEE International Conference on Systems. Man Cybern 3:2153–2158
10. Bai H, Zhu J, Liu C (2003) A fast license plate extraction method on complex background, Proceedings of the 2003 IEEE International Conference on Intelligent Transportation Systems. Vol. 2:985–987
11. Suwa M, Wu Y, Kobayashi M, Kimachi M, Ogata S (2000) A stereo based vehicle detection method under windy conditions In Proceedings of the IEEE Intelligent Vehicles Symposium 2000 (Cat. No. 00TH8511), pp. 246–248
12. Yoshimori S, Mitsukure Y (2003) License plate detection system in rainy days, Proceedings 2003 IEEE International Symposium on Computational Intelligence in Robotics and Automation. Computational Intelligence in Robotics and Automation for the New Millennium (Cat. No. 03EX694). Vol. 2. IEEE, 2003, pp.972–976
13. Panahi R, Gholampour I (2017) Accurate detection and recognition of dirty vehicle plate numbers for high-speed applications. IEEE Trans Intell Transp Syst 18(4):767–779
14. Yuan Y, Zou W, Zhao Y, Wang X, Hu X, Komodakis N (2017) A robust and efficient approach to license plate detection. IEEE Trans Image Process 26(3):1102–1114
15. Azam S, Islam MM (2016) Automatic license plate detection in hazardous condition. J Vis Commun Image Represent 36:172–186
16. Hsu GS, Ambikapathi A, Chung SL, Su CP (2017) Robust license plate detection in the wild, in 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp.1–6
17. Wu Y, Liu S, Wang X (2013) License plate location method based on texture and color, 2013 IEEE 4th International Conference on Software Engineering and Service Science. IEEE, pp. 361–364
18. Musoromy Z, Ramalingam S, Bekooy N (2010) Edge detection comparison for license plate detection, 2010 11th International Conference on Control Automation Robotics & Vision. IEEE 2010:1133–1138
19. Yang F, Ma Z (2005) Vehicle license plate location based on histogramming and mathematical morphology, Fourth IEEE Workshop on Automatic Identification Advanced Technologies (autoid'05). IEEE 2005:89–94
20. Babu C, Kennedy N, Nallaperumal K (2008) A license plate localization using morphology and recognition, 2008 Annual IEEE India Conference. Vol. 1. IEEE, 2008, pp.34–39
21. Gou C, Wang K, Yao Y, Li Z (2016) Vehicle license plate recognition based on extremal regions and restricted boltzmann machines. IEEE Trans Intell Transp Syst 17(4):1096–1107
22. Burges CJC (1998) A tutorial on support vector machines for pattern recognition. Data Min Knowl Disc 2(2):121–167
23. Belhumeur PN, Hespanha JP, Kriegman DJ (1997) Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. IEEE Trans Pattern Anal Mach Intell 19(7):711–720
24. Boser BE, Guyon IM, Vapnik VN (1992) A training algorithm for optimal margin classifier, In Proceedings of the fifth annual workshop on Computational learning theory, pp.144–152
25. Mazanec J, Melisek M, Oravec M, Pavlovicova J (2008) Support vector machines, PCA and LDA in face recognition. J Electr Eng 59(4):203–209
26. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You Only Look Once: unified, real-time object detection, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.779–788
27. Ren S, He K, Girshick R, Sun J (2017) Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell 39(6):1137–1149

28. Li H, Shen C (2016) Reading car license plates using deep convolutional neural networks and LSTMS. arXiv preprint arXiv: 1601.05610
29. Montazoli S, Jung C (2017) Real-time Brazilian license plate detection and recognition using deep convolution neural networks, 2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI). IEEE, pp. 55–62
30. Rafique MA, Pedrycz W, Jeon M (2017) Vehicle license plate detection using region-based convolutional neural networks. Soft Comput 22(19):6429–6440
31. Bulan O, Kozitsky V, Ramesh P, Shreve M (2017) Segmentation and annotation-free license plate recognition with deep localization and failure identification. IEEE Trans Intell Transp Syst 18(9): 2351–2363
32. Yonetsu S, Iwanmato Y, Wei Chen Y (2019) Two-stage YOLOv2 for accurate license-plate detection in complex scenes, 2019 IEEE International Conference on Consumer Electronics (ICCE), pp1–4
33. Redmon J, Farhadi A (2017) YOLO9000: better, faster, stronger, Proceedings of the IEEE conference on computer vision and pattern recognition, pp.7263–7271
34. Laroca R, Severo E, Zanlorensi LA, Oliveira LS, Gonçalves GR, Schwartz WR, Menotti D (2018) A robust real-time automatic license plate recognition based on the YOLO detector, 2018 International Joint Conference on Neural Networks (IJCNN), pp1–10