License Plate Localization in Complex Environments Based on Improved GrabCut Algorithm

HENGLIANG SHI1 AND DONGNAN ZHAO2
1Automobile and Rail Transit School, Luoyang Polytechnic, Yibin, Luoyang 471023, China
2Information Engineering School, Henan University of Science and Technology, Luolong, Luoyang 471000, China

ABSTRACT Aiming at the problem that the existed license plate detection method lacking of accuracy and speed, an improved lightweight detection algorithm for license plate detection in natural scenarios was proposed. First, the traditional GrabCut algorithm needs to interactively provide a candidate frame in order to perform the target detection work. We replace the candidate frame by introducing the Aspect ratio of the license plate as the foreground extraction feature to automate the detection of the license plate by GrabCut algorithm. Then, in order to improve the detection precision of traditional target detection algorithms, we introduced the Wiener filter, which is widely used in the field of digital signal processing, and combine with Bernsen algorithm to complete image noise reduction. Finally, the algorithm was tested with the CCPD dataset, which contains many vehicle images from different complex natural scenes, especially low-resolution images. The experimental results shows that improved GrabCut algorithm achieves an average accuracy of 99.34% for license plate localization and a detection speed of 0.29s/frame, which has better accuracy and real-time performance compared with traditional GrabCut and other license plate localization algorithms.

INDEX TERMS Aspects ratio, GrabCut, license plate location, object detection, Wiener filter.

I. INTRODUCTION
License plate location recognition is a key technology in the intelligent transportation system, and the realization of high-precision automatic license plate location recognition under all-weather natural conditions is of great significance to road traffic safety monitoring, traffic congestion alleviation, environmental pollution reduction and other social problems. As a prerequisite step of License Plate Recognition (LPR) technology, the efficiency and accuracy of license plate positioning directly affects the effectiveness of license plate recognition. License plate localization methods are mainly classified into methods based on a priori information and detection methods using deep learning models.

License plate localization methods based on a priori information are generally classified as color texture [1], [2], shape regression [3], [4], and edge detection [5], [6]. The color texture-based approach focuses on determining the space that matches the license plate texture by comparing each pixel point in two grayscale images and calculating the license plate area, which is generated from the background and character colors of the license plate in the image. Tang and Zhang [1] extracted color information based on the license plate features of domestic cars with white characters on a blue background, and located the candidate area of approximate blue license plates based on the RGB ratio of color images, but they could not locate the license plates of blue cars; Ling and Huang [3] according to the difference between the license plate characters and the background in the binarized image to produce grayscale jump, coarsely locate the line position of the license plate in the image and narrow the finding range for license plate location. Finally, the color features and shape features are used to precisely locate all license plates in the coarse-located small range of images. This method can locate...
multiple license plates effectively, but it is easy to lead to false detection for license plate smudging, aging and deformation; Li [5] uses Sobel to detect the edge information of license plates in images and achieves license plate localization by selecting and combining candidate regions, making full use of shape and texture features to reasonably determine thresholds and develop strategies. The success rate of license plate localization by this method can reach more than 90%. However, for cases such as blurring or too many borders or too large images during the shooting process, the edge information and texture features of the license plate will be lost, which then affects the license plate localization accuracy of the method.

Most of the above traditional license plate localization methods need to give a priori features manually and interactively, which can achieve good localization results under certain circumstances such as normal lighting, clear images, and small plate angle tilt. However, the localization accuracy in unconstrained environment is average, mainly because the feature information used in traditional localization methods is less adaptable, and the license plate images collected in unconstrained environment are seriously affected by various composite factors, so more adaptable feature information needs to be considered for its precise localization.

Current domestic and international research on license plate location detection mainly uses methods based on machine learning or deep learning. Machine learning-based license plate localization methods mainly include SVM classifier [7], BP neural network [8] and AdaBoost classifier [9], etc. These algorithms are used to extract features, train algorithmic models on a large set of image samples provided in advance to obtain classifiers, and then use the classifiers to localize the images in the test set. This type of algorithm model is robust and has a certain effect on the localization of smudged, covered and deformed license plates. However, over-reliance on features to construct sample datasets makes training difficult and time-consuming. There are two main approaches for license plate localization by deep learning, one is the two-stage model represented by RCNN, such as RCNN [10], SPP-NET [11], Fast R-CNN [12], and Faster R-CNN [13], and the other is the one-stage (one-stage) model detection algorithm represented by YOLO [14] series and SSD [15], which use one CNN network to directly predict the class and location of different objects. The above algorithms perform well in license plate location recognition, but various network models require building a huge network structure and training a large sample data set, which has a large overhead in practical applications and the algorithms run too slowly.

Generally, the images for license plate positioning are collected in real time by cameras at intersections or parking lots, as shown in Fig. 1, and the vehicle images contain a large amount of noise. In order to make the images more compatible with computer vision processing, we need to perform pre-processing operations on the collected images to eliminate noise and enhance the image quality. Image enhancement can improve the visual effect of an image by purposefully highlighting general or local features of the image and suppressing uninteresting features, and is one of the extremely effective pre-processing methods. Image enhancement is mostly classified into spatial domain method and frequency domain rate method. The spatial domain method is a straightforward image enhancement algorithm which includes gray level calibration, contrast stretching and histogram correction. This algorithm can achieve idealized distribution of image grayscale and contrast enhancement, but it will reduce the local details of the image when merging numerous gray levels and is not suitable for small target localization detection. Frequency domain method is an indirect image enhancement algorithm and frequently used frequency domain enhancement methods are low-pass filter and high-pass filter. They treat an image as a two-dimensional signal and perform Fourier transform-based signal enhancement on it. Low-pass filtering removes the noise in the image, and high-pass filtering enhances high-frequency signals such as edge information to make the blurred part of the image clear. However, this method will lose part of the detail information in the image while enhancing the image.

To address the above problems, this paper proposes a high-precision, lightweight license plate localization algorithm. The main contributions of this paper are as follows:

- We abandon the semi-automatic mode of traditional GrabCut algorithm which requires users to give a priori features interactively, and introduce the Aspect ratio of license plate as a rectangular feature attribute to initialize GrabCut algorithm to achieve automatic localization of license plate.
- We propose a K-value adaptive estimation method, which improves the automatic processing capability and noise reduction effect of Wiener filtering. And combined with Otsu binary method, the rectangular feature of license plate has the symbolic, which improves the accuracy and efficiency of license plate localization.

II. ORIGINAL ALGORITHM

GrabCut algorithm was first developed by Rother et al. in 2004 as an iterative graph segmentation algorithm that features a reduced number of manual interactions for image segmentation [16], which uses texture (color) and boundary (contrast) information in an image to perform efficient segmentation and is a typical foreground and background segmentation algorithm. It is based on Graph Cut algorithm, which requires the user to set the seed points of the target and background, and then obtain the segmentation result by just one calculation of the energy formula, which is usually not effective. GrabCut algorithm solves this problem by changing the modeling and segmentation methods from a grayscale histogram model of the target and background.
to RGB three-channel hybrid Gaussian model GMM. And the energy equation is calculated interactively and iteratively using continuous partition estimation and model parameter learning. In the interaction phase GrabCut only requires the user to box out the target, then all pixels outside the box will become background, i.e. GrabCut algorithm allows incomplete annotation. The graph segmentation process of GrabCut is shown in Fig. 2.

First, make a rectangular box in the input image, and any part outside the rectangle is the background. The PC will make an initial marker of our input image, marking the foreground and background pixels. Afterwards, a Gaussian Mixture Model (GMM) is used to model the foreground and background. The GMM learns and creates new pixel distributions based on our input. For contested pixel parts of the image (foreground or background), GMM will do clustering operations based on their relationship with known classified pixels. From the above process, we will get an image created from the pixel distribution, where all nodes are pixel points and there are two nodes in addition to the pixel points: Source_node and Terminal_node. The foreground pixels are connected to Source_node and the background pixels are all connected to Terminal_node. At this point, we will get the cost equation based on the weight information of Source_node and Terminal_node, as shown in equation (1):

$$E(K) = \theta U(K) + V(K)$$ (1)

In equation (1), $E$ denotes the energy, $U$ is the area term that represents the energy value required to cut the edge S-links, and $V$ is the boundary term that represents the energy value required to cut the edge T-links. $\theta \in (0, 1)$, taking a value of 0 for the background component and a value of 1 for the foreground component.

### III. IMPROVED ALGORITHM

The traditional GrabCut algorithm requires the user to perform certain editing steps to complete the input work. In order to make the automation of the algorithm process possible, scholars at home and abroad have given different attempted methods in recent years. Zhou et al. proposed to use part of the algorithm’s detection results as preprocessing information, and this method achieves semi-automatic operation of the algorithm [17]. Khattab et al. automated GrabCut using clustering techniques Kmeans and Fuzzy C-means [18], but the introduction of clustering algorithms significantly reduced the operational efficiency of the GrabCut algorithm. Salau et al. introduced the method of shape features of license plates as a priori features of GrabCut to achieve full automation of GrabCut [19]. However, this method does not conduct further research on license plate localization, and the noise reduction problem of high noise images in complex environments is relatively single consideration. We will perform optimization work on the algorithm itself as well as on the image pre-processing stage to achieve a localized extension of this license plate localization algorithm.

GrabCut algorithm starts the image segmentation by first converting the input image into a trilateration (Trimap). Trimap is a common annotation method that will perform a kind of rough division of a given image into foreground, background and unknown areas. In this paper, we set the value of the license plate Aspect ratio $T$ based on the current national standard size of China’s nine-two type motor vehicle number plates. As shown in the Fig. 3. The size of the national standard blue and black plates is $440 \times 140$, and the size of the front plate of the yellow plate is the same as above, and the rear is $440 \times 220$, all in mm. Considering the effect of slight dumping and defacement of license plate and the variability of license plate size after yellow plate, we get the threshold range of Aspect ratio $T$ according to equation (2) is $1.9 \leq T \leq 3.4$. $T$ will be used as an a priori feature attribute to get the initialized Trimap, which does not require the user to complete the boxing of the rectangular region manually, successfully automating the work of GrabCut algorithm.

$$\text{new}_h, \text{new}_w = \text{self}.\text{output}_\text{size}, \text{self}.\text{output}_\text{size}$$
$$\times \left(\frac{w}{h}\right) \quad (w > h)$$ (2)

In the resulting Trimap, we mark the part of the rectangle within the Aspect ratio $T$ as the foreground and the rest as the background. The foreground and background data are modeled using a Gaussian mixture model to achieve a minimal partition of the energy function. The Gaussian mixture model can adapt to more complex and variable application scenarios, with better fitting and computational performance. It consists of the weighted average of $K$ single Gaussian components, and the probability distribution of each Gaussian component is expressed as follows.

$$N(x; \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^d|\Sigma|}} \exp \left[-\frac{1}{2}(x - \mu)^T \Sigma^{-1}(x - \mu)\right]$$ (3)

In equation (3), $x$ is a d-dimensional vector, i.e., multidimensional data, obeying a Gaussian distribution with mean $\mu$.
and variance $\Sigma$. Assuming that the image is partitioned into $K$ parts, each of which obeys a Gaussian distribution with mean $\mu_k$ and variance $\Sigma_k$, i.e., obeys $N(\mu_k, \Sigma_k)$, the probability density function of the Gaussian mixture model is:

$$p(x; \mu, \Sigma) = \sum_{k=1}^{K} w_k \cdot N(x; \mu_k, \Sigma_k)$$

(4)

In equation (4), $w_k, \mu_k, \Sigma_k$ is the weighted proportion of the $k$-th Gaussian distribution in the model. To ensure the non-negativity and normalization properties of the probability density function $p(x)$, $w_k$ must satisfy the condition that:

$$0 \leq w_k \leq 1, \sum_{k=1}^{K} w_k = 1$$

(5)

From equation (4), it is known that the Gaussian mixture model has three parameters to be solved, which are denoted by $\theta$ here, $w_k, \mu_k$, and $\Sigma_k$. Using the idea of maximizing the log-likelihood function, the optimal values of all parameters can be solved to obtain the best clustering results and the new pixel distribution map.

To perform the GrabCut algorithm, the pixel distribution graph is converted into a graph consisting of vertices and edges. Since we introduce the Aspect ratio $T$ of the Chinese license plate as an a priori rectangular feature, substitution into the energy equation (1):

$$E(K, T) = \theta U(K, T) + V(K, T)$$

(6)

The area term $U$ in Equation (6) is defined as follows:

$$U(K, T) = \sum R(K_n, T_n) R(K, T)$$

(7)

For the function $R$ in the region term $U$, it denotes the independent penalty of the pixel point $K_n$ labeled as foreground or background for a given grayscale histogram model in Graphcut, also known as the degree to which the pixel point $K_n$ fits into the model. Parameters $\theta$ in equation (6) is the weighting factor between foreground and background, and when $\theta$ or $U(K, T) = 0$, we know:

$$E(K, T) = V(K, T)$$

(8)

As can be seen from equation (8), when $T$ is introduced, we can locate the boundary between the foreground and background and extract $K_n$ while obtaining the minimum value of the energy function $E$.

The regional term $U$ can be found by bringing equation (4) into the energy function equation (6):

$$U(\alpha, K, \theta, Z) = -\log w(\alpha_n, k_n) + \frac{1}{2} \log \left| \sum (\alpha_n, k_n) \right|$$

$$= \log w(\alpha_n, k_n) + \frac{1}{2} [z_n - \mu(\alpha_n, k_n)]^T$$

$$\times \sum (\alpha_n, k_n)^{-1} [z_n - \mu(\alpha_n, k_n)]$$

(9)

The boundary term $V$ is the same as GraphCut, which also represents the continuity between adjacent pixel points, and usually uses the Euclidean distance measure of similarity, so the boundary term formula is:

$$V(\alpha, Z) = \gamma \sum_{(m,n) \in C} \|z_m - z_n\|^2$$

(10)

where $\|z_m - z_n\|$ is the Euclidean distance of pixel points, $\gamma$ is the adaptive parameter, and $\beta$ is a constant. The Grabcut algorithm achieves the best separation when $\gamma = 50$, while the value of $\beta$ is calculated based on the contrast of the image to be processed. After a large number of experimental studies, the results show that if the image contrast is high, the value of $\beta$ is relatively large, otherwise the value of $\beta$ has to be relatively small.

Finally, the GrabCut algorithm will keep iterating to make the energy function $E$ in equation (6) decreasing, and the best localization separation of images will be achieved when the minimum value is reached. We organize the above steps of the algorithm formula and present the improved GrabCut Chinese license plate localization algorithm in the form of pseudo-code, as shown in Algorithm 1.

**Algorithm 1 Improved GrabCut Chinese License Plate Location Algorithm**

**Require:**
- Color vehicle images;
- contains a color image of the vehicle;

**Ensure:**
- Sub-images containing license plate positioning areas;
- Input contains a color image of the vehicle;
- Normalized image size;
- Wiener filter de-noising;
- Create a Trimap (1.9 ≤ Ar ≤ 3.4);
- Label image foreground(fg), background(bg), pixels(p);
- Modeling of fg and bg using GMM;
- Pixel distribution to build images(Vertices and edges);
- Grayscale(Weighted average method);
- Binarization(Otsu binary method);
- Execute license plate location;
- Save sub-images to local files;
- //GrabCut optimized algorithm license plate location completed;

**Return** Sub-images with license plate positioning area;
As shown in the figure, a-d indicate the results of binary of the original images in different environments using the Otsu binary methods. The high noise problem causes weak contrast between the license plate area and the background area in the binaries images, and the license plate parts in Fig. 4 are even between the license plate area and the background area in the binary methods. The high noise problem causes weak contrast the original images in different environments using the Otsu binarization.

In this paper, we use improved Wiener filtering algorithm to perform noise reduction and Otsu binary methods on the four types of images in Fig. 5. As shown in Fig.5(a-b), it can be clearly seen from the figure that the image after noise reduction using the improved Wiener filtering algorithm is sharper than the grayscale map of the original image. In the binary image, the license plate region has higher contrast with the background region, and the rectangular features of the license plate are more obvious without missing and corrupted phenomena, which can be better adapted to the training of the improved GrabCut algorithm. The improvement of the K-value adaptive estimation method enhances the automatic processing capability and noise reduction effect of the Wiener filter, which makes the Wiener filter be more widely used in image de-noising work.

### B. GRAY-SCALE CONVERSION AND BINARIZATION

In this section, we will give a detailed description of the grayscale conversion and binarization processes in the previous image preprocessing process.

The input original vehicle images are 24-bit RGB images. In this paper, the grayscale calculation method with the weighted average of equation (17) is used to convert the color
Let the threshold be \( t \), and divide the grayscale image into computational accuracy will become higher.

The colors of Chinese license plates are all yellow and blue, and the grayscale difference with the background part is obvious, so we choose Otsu binary methods. The method uses the variance of the uniformity of the grayscale distribution as the unit of measure, and the larger the variance value, the greater the difference between the two parts that make up the image. When the target region (license plate) is missegmented from the background (vehicle or scene), it leads to a smaller inter-class variance, so a segmentation threshold that maximizes the between-class variance means that the computational accuracy will become higher.

We set the number of pixels with grayscale value \( i \) to be \( a_i \). The total pixel value \( A = \sum_{i=0}^{l-1} a_i \) of the grayscale image, and thus the probability of occurrence of a pixel with grayscale value \( i \), is \( p_i = a_i / A \), where \( p_i \geq 0 \) and \( \sum_{i=0}^{l-1} p_i = 1 \). Let the threshold be \( t \), and divide the grayscale image into background \( W_1 \) and foreground \( W_2 \), i.e., \( W_1 : \{0, 1, 2 \cdots t \} \), \( W_2 : \{t + 1, t + 2, t + 3 \cdots l - 1 \} \), the probabilities of \( W_1 \) and \( W_2 \) appearing:

\[
p_1(t) = \sum_{i=0}^{t} p_i \tag{18}
\]

\[
p_2(t) = \sum_{i=t+1}^{l-1} p_i \tag{19}
\]

The mean grayscale values of \( W_1 \) and \( W_2 \):

\[
\mu_1(t) = \sum_{i=0}^{t} i p_i / p_1(t) \tag{20}
\]

\[
\mu_2(t) = \sum_{i=t+1}^{l-1} i p_i / p_2(t) \tag{21}
\]

The overall grayscale mean of the image:

\[
\mu = \sum_{i=0}^{l-1} i p_i \tag{22}
\]

The between-class variance of \( W_1 \) and \( W_2 \) can be calculated:

\[
\sigma^2(t) = p_1(t) |\mu_1(t) - \mu(t)|^2 + p_2(t) |\mu_2(t) - \mu(t)|^2 \tag{23}
\]

A larger between-class variance indicates a larger gray difference between \( W_1 \) and \( W_2 \), and a better segmentation of the license plate region from the background region. So the core idea of Otsu algorithm is in finding the maximum value of the inter-class variance between foreground and background regions, which is to confirm the best segmentation threshold. The calculation formula:

\[
t^* = \arg \max_{0 < t < l - 1} \sigma^2(t) \tag{24}
\]

From equation (24) we can find the optimal threshold and generate binary images. The process of grayscale conversion and binarization is done by python programming language, and the result of the process is shown in Fig. 6(a-b). From left to right, figure a represents the original input image; figure b represents the grayscale image; figure c represents the grayscale histogram of the image; figure d represents the obtained binary image, from which we can see that the features of the license plate are well-preserved and the complexity of the image has been significantly decreased.

C. RESULTS AND DISCUSSION

1) LICENSE PLATE POSITIONING PROCESS AND RESULTS

In this section, we give the procedure and experimental results of the improved GrabCut algorithm license plate localization, briefly introduce the improvement of the algorithm source code, and compare our method with the existing methods. We use open CV+python in Linux-based system environment for training and testing, and choose the improved GarbCut algorithm for license plate location detection. The GPU is NVIDIA GeForce GTX 1060Ti and the memory is 16G. The license plate dataset used for the experiments in this paper is the open source large-scale Chinese urban parking dataset CCPD [21] and some vehicle data images collected by ourselves, totaling 7500 images. The images in the dataset are all \( 1160 \times 720 \) in size and contain images from many complex environments, such as bright light, cloudy sky, dark light, smudged license plates, tilted, etc., constituting a dataset with a diverse and sufficient number of license plate scenes distributed as shown in Table 1.

We modified the source code of the GrabCut algorithm in the OpenCV library to automate the GrabCut algorithm by using the license plate Aspect ratio \( T \) initialization algorithm, which is done by placing if statements to complete check the validity of the input. In order to see the effect of license plate positioning more visually, we instead use a red rectangular
TABLE 1. Composition of CCPD dataset.

| CCPD   | Numbers/k | Description               |
|--------|-----------|---------------------------|
| Base   | 200       | Normal license plate       |
| FN     | 20        | Too far or too close      |
| DB     | 20        | Light is too dark or too bright |
| Rotate | 10        | Level(20-23°),Vertical(10-10°) |
| Tilt   | 10        | Level(15-45°),Vertical(15-45°) |
| Weather| 10        | Rainy,snowy or foggy days |
| Blue   | 5         | Blue caused by camera shake |
| Challenge | 10    | Other challenging Lp      |
| NP     | 5         | New cars without Lp       |

FIGURE 7. License plate location with improved GrabCut.

The detection effect of the improved GrabCut algorithm on license plate under unconstrained environment is shown in Fig. 7. Different complex environments such as fog, night, blurred, dumped license plate, and specific positioning effects of different license plate types such as blue car license plate and yellow truck license plate are listed in the figure. From the figure, we can intuitively see that the improved algorithm proposed in this paper has better localization effect on license plates in various complex scenes, which fully proves that the proposed algorithm improvement scheme is reasonable and feasible. In order to verify the superiority of the improved algorithm, we use the traditional GrabCut algorithm to work on locating the license plates in Figure 1, and the results are shown in Fig. 8, where half of the license plates have failed to locate. As we can see from a and c, due to the lack of good noise reduction work on the original image, the rectangular features of the license plate in the resulting binary image are missing, which makes the final license plate positioning deviate, but this deviation often makes the positioned license plate image missing a certain license plate number, which causes great trouble to the subsequent license plate recognition work. As we can see in Fig. b and Fig. d, the low quality raw image directly makes the license plate features in the generated binary image disappear completely or blend in with the surrounding background, resulting in a complete failure of localization.

2) ACQUISITION OF EDGE WEIGHTS

After data enhancement of the original image, the edge weight values of the boundary terms in the generated GrabCut image will become smaller, which means that the improved GrabCut algorithm pays less in minimizing the energy function |E| and the image segmentation is better and faster. We compute the edge weights for some GrabCut images with the original image, and represent the edge weights of the images logarithmically on the vertical coordinates and the labels of the images on the horizontal coordinates (1-15). The results of comparing the edge weight values of GrabCut images and the original images are shown in Fig. 10, which indicates that for all extracted test images, the edge term weight values of GrabCut images are smaller than those of the original images.

3) COMPARISON OF PRECISION AND EFFICIENCY

To verify the accuracy of Improved GrabCut for batch automated license plate localization of vehicle images, we conduct license plate localization experiments using Improved GrabCut and traditional GrabCut for different environment datasets, respectively. The statistics of license plate location results for both algorithms are shown in Fig. 11. The horizontal coordinates in the figure represent the type of complex
The GrabCut algorithm improved for automatic target detection in this paper has improved license plate localization accuracy compared with the original algorithm. However, considering that the higher positioning accuracy of traditional GrabCut comes more from the manual accurate foreground framing, which has a great impact on the accuracy of positioning segmentation, the improved GrabCut changes this situation and ensures the accuracy of positioning while automating GrabCut algorithm.

As shown in the Table (2), the license plate detection method with improved GrabCut + Wf + Otsu in this paper has the highest accuracy of recognition compared with the original or partially improved method, and the overall performance is optimal despite the increased detection process and complexity and the increase in time. In addition, when the improved Wiener filter is used for image noise reduction, the accuracy of license plate detection is improved by about 6%, proving the necessity and effectiveness of using this method.

To quantitatively evaluate the performance of the improved GrabCut algorithm, we selected six commonly used license plate localization algorithms to perform license plate localization experiments on vehicle images in a dataset together with Improved GrabCut, using a dataset that includes a test set and six subsets of data in a single environment except for the Base subset. We counted the license plate location accuracy of various algorithms and also calculated the average accuracy AP of the algorithms in the whole test set, as shown in Table (3).

As can be seen from Table (3), the license plate localization accuracy of the algorithm in this paper is higher than that of the other six algorithms tested in the literature, i.e., the first six algorithms in Table (3), when compared with the license plate localization accuracy of the other algorithms in the literature queried, both in the Base subset and in the other six single environments detected.

Meanwhile, the accuracy of the algorithm in this paper reaches more than 99.00% on Tilt and Weather subsets, which verifies the effectiveness of the algorithm in this paper. We can see that the license plate localization algorithms generally perform poorly on the Fn and DB subsets, and the next step can be to optimize the performance of the algorithms starting from two single scenarios, Fn and DB. We can visualize from the license plate location time of the algorithms in Table (3) that the lightweight graphical segmentation algorithm runs much more efficiently than the license plate location algorithm that builds a neural network framework and is suitable as the core algorithm for embedded license plate location devices.

4) THRESHOLD SELECTION
To further evaluate the impact of the Aspect ratio thresholds on the improved GrabCut algorithm, the accuracy of the algorithm under different Aspect ratio thresholds is statistically evaluated in this paper, as shown in Table (4).

| Model                | Precision(%) | Time(ms) |
|----------------------|--------------|----------|
| GrabCut              | 87.26        | 52       |
| GrabCut + Otsu       | 91.41        | 54       |
| GrabCut + Wf + Otsu  | 97.02        | 65       |
| Im GrabCut + Otsu    | 93.53        | 50       |
| Im GrabCut + Wf + Otsu| 99.34        | 56       |

| Algorithm              | AP  | P(r) | t(s) | Ba  | Fn   | DB   | Rotate | Tilt | Weather | Blur | Challenge |
|------------------------|-----|------|------|-----|------|------|--------|------|---------|------|------------|
| Faster R-CNN[21]       | 92.90 | 0.72 | 98.10 | 83.70 | 92.10 | 91.80 | 89.40 | 81.80 | 83.90   |
| YOLO-9000[22]         | 93.10 | 0.85 | 93.10 | 77.30 | 89.60 | 93.30 | 91.80 | 84.20 | 88.60   |
| SSD-300[23]           | 94.40 | 0.69 | 99.10 | 84.70 | 89.20 | 95.60 | 94.90 | 85.40 | 93.10   |
| T2E2[24]              | 94.20 | 1.12 | 98.50 | 83.80 | 91.70 | 95.10 | 94.50 | 83.60 | 93.10   |
| YOLO-v3[25]           | 99.13 | 1.26 | 99.94 | 91.04 | 94.35 | 97.87 | 99.00 | 99.73 | 97.74   |
| RNet[21]              | 94.50 | 0.56 | 99.30 | 85.30 | 89.50 | 94.70 | 93.20 | 84.10 | 92.80   |
| Im-GrabCut            | 99.19 | 0.31 | 99.96 | 93.09 | 95.34 | 98.42 | 99.29 | 99.71 | 97.76   |

| Ar       | AP  | Base | Fn   | DB   | Rotate | Tilt | Weather | Blur |
|----------|-----|------|------|------|--------|------|---------|------|
| 1.90     | 91.74 | 93.72 | 89.72 | 90.72 | 89.32  | 87.98 | 87.64   | 90.48 |
| 2.40     | 95.23 | 96.32 | 94.22 | 93.21 | 91.45  | 92.32 | 94.19   | 93.97 |
| 2.60     | 98.46 | 96.21 | 96.32 | 97.72 | 97.21  | 96.96 | 97.54   | 99.21 |
| 3.00     | 97.32 | 98.21 | 96.62 | 96.78 | 97.71  | 97.86 | 96.32   | 96.76 |
| 3.40     | 92.72 | 93.86 | 92.29 | 91.17 | 92.21  | 92.45 | 91.79   | 92.01 |
From Table (3), we can see that when the Aspect ratio threshold is in the range of 2.10 – 3.2, the algorithm in this paper still has a significant advantage over the other algorithms in Table (2) in terms of positioning accuracy. At the threshold values of 1.90 and 3.40, the accuracy of license plate positioning decreases more obviously, mainly because the accuracy requirement of algorithm positioning becomes higher when the threshold value is set too small or large. In general, the accuracy of threshold prior information in target detection is also only relative, so the algorithm detection allows a certain error between the detected target area and the actual labeled area.

V. CONCLUSION

In this paper, we propose an improved GrabCut Chinese license plate location detection algorithm, which automates the operation of the algorithm by introducing a license plate Aspect ratio threshold while ensuring the accuracy of license plate location. In order to solve the problem of difficult and inefficient license plate location in complex environments, this paper adopts an improved adaptive Wiener filtering algorithm for image enhancement, and reduces the complexity of the image by the operation of graying and Otsu binary methods to achieve high accuracy localization of license plates by the algorithm while ensuring detection efficiency. We test the algorithm of this paper on the CCPD dataset, and the experimental results show that the algorithm can effectively improve the accuracy and efficiency of license plate location detection in complex environments, and its average accuracy reaches 99.34%, corresponding to a processing time of 0.29s.

In future work, we will add the character recognition function of license plates to the algorithm, so that the method can be used as a lightweight license plate location recognition algorithm for embedded devices in “smart transportation”.

REFERENCES

[1] Y. Tang and S. Zhang, “Color-based license plate recognition research,” Mod. Comput., vol. 32, pp. 63–66 and 71, 2020.

[2] X. Li, “Extraction of Macau car license plate location based on image texture recognition,” Inf. Technol. Informatication, vol. 11, pp. 237–240, Jan. 2020.

[3] X. Wang and B. Huang, “Multi-license plate location based on improved two-dimensional discrete wavelet transform,” J. ChongQing Jiaotong Univ., Natural Sci., vol. 39, no. 2, pp. 16–21, 2020.

[4] Z. Zhang and Y. Su, “Location method for license plate based on M-HSI and pixel offset,” Comput. Eng. Des., vol. 39, no. 11, pp. 3576–3583, 2018.

[5] X. Li, “Location extraction of Macau vehicle license plate based on image texture recognition,” Inf. Technol. Informatication, vol. 11, pp. 237–240, 2020.

[6] M. Ravichandran and S. Sumithra, “Robust automated license plate and character recognition,” Diet. Image Process., 2012.

[7] K. Gao, H. Jia, and X. Guo, “Application of CNN inlicense plate classifier,” Comput. Eng. Appl., vol. 53, pp. 209–213, 2017.

[8] Y. Gao, K. Wang, Y. Yao, and Z. Li, “Vehicle license plate recognition based on extremal regions and restricted Boltzmann machines,” IEEE Trans. Intell. Transp. Syst., vol. 17, no. 4, pp. 1096–1107, Apr. 2016.

[9] M. S. Al-Shemarry, Y. Li, and S. Abdulla, “Ensemble of AdaBoost cascades of 3L-LBPv classifiers for license plates detection with low quality images,” Expert Syst. Appl., vol. 92, pp. 216–235, Feb. 2018.

[10] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part based models,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[11] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 37, no. 9, pp. 1904–1916, Sep. 2014.

[12] G. Shi, “Fast R-CNN,” in Proc. IEEE Int. Conf. Comput. Vis., Dec. 2015, pp. 1440–1448.

[13] W. Li, “Analysis of object detection performance based on faster R-CNN,” J. Phys., Conf., vol. 1827, no. 1, Mar. 2021, Art. no. 012085.

[14] Y. Wang, C. Qi, and Q. Du, “Target recognition of heterogeneous fusion images based on improved YOLO model deep learning algorithm,” J. Phys., Conf., vol. 2170, no. 1, Feb. 2022, Art. no. 012011.

[15] T. Szarvas and Z. Kis, “Numerical simulation of nonlinear second harmonic wave generation by the finite difference frequency domain method,” J. Opt. Soc. Amer. B, Opt. Phys., vol. 35, no. 4, p. 731, 2018.

[16] C. Rother, V. Kolmogorov, and A. Blake, “GrabCut interactive foreground extraction using iterated graph cuts,” ACM Trans., vol. 23, no. 3, pp. 309–314, 2004.

[17] X. Zhou, T. Ito, X. Zhou, and H. Chen, “A universal approach for automatic organ segmentations on 3D CT images based on organ localization and 3D GrabCut,” Proc. SPIE, vol. 9035, pp. 745–752, Mar. 2014.

[18] D. Khatib, H. M. Ebeid, M. F. Tolba, and A. S. Hussein, “Clustering-based image segmentation using automatic GrabCut,” in Proc. 10th Int. Conf. Informat. Syst. (INFOS), 2016, pp. 95–100.

[19] A. O. Salau, T. K. Yesufu, and B. S. Ogundare, “Vehicle plate number localization using a modified grabcut algorithm,” J. King Saud Univ. Comput. Inf. Sci., 2019.

[20] M. Qiao and G. Sun, “Speed-limit traffic road signs detection based on Wiener restoration,” Softw. Guide, vol. 19, pp. 234–237, 2020.

[21] 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,” in Proc. Eur. Conf. Comput. Vis. Heidelberg, Germany: Springer, 2018, pp. 261–277.

[22] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards realtime object detection with region proposal networks,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[23] J. Redmon and A. Farhadi, “YOLOv3: Better, faster, stronger,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 6517–6525.

[24] W. Liu, D. Anguelov, and D. Erhan, “SSD: Single shot multibox detector,” in Proc. Eur. Conf. Comput. Vis. Heidelberg, Germany: Springer, 2016, pp. 21–37.

[25] H. Li, P. Wang, and C. Shen, “Toward end-to-end car license plate detection and recognition with deep neural networks,” IEEE Trans. Intell. Transp. Syst., vol. 20, no. 3, pp. 1126–1136, Mar. 2019.

[26] J. Redmon and A. Farhadi, “YOLOv3: An incremental improvement,” 2018, arXiv:1804.02767.