MSER and Machine Learning-based License Plate Recognition

Linghao Zeng
International School
Jinan University
Guangzhou, China
linghao.z@foxmail.com

***Abstract*—This paper is an undergraduate project for the Digital Image Processing course. A MSER (Maximally Stable Extremal Region) and machine learning-based method is proposed in this paper for the software part of vehicle license plate recognition. First, we locate the license plate by color features, and tilt correct and pre-process the plate by graphical and morphological methods, and then, the candidate MSER license plate regions are chosen according to the pixel sum, the scale and length-width ratio of the license plate character region, finally, the characters are segmented according to MSER regions and passed into the trained neural network for recognition. A GUI for operation is also provided in this paper.**

Keywords—License place, MSER, deep learning, GUI

I. Introduction

The recognition of license plate is a important technique in today's lives. A complete license plate recognition system completes the process from image acquisition to character recognition output, which is quite complex and can be divided into hardware and software parts. In most cases, the software part includes image pre-processing, license plate positioning, character segmentation, and character recognition.

This paper addresses the great advantages of MSER for detection character regions. In previous work, MSER may not have been the most mainstream approach [1], but it has its own strengths in complex nature scenes to extract character features, which can reduce image pre-processing workload (we know that for small devices, arithmetic resources are valuable). Using neural network for character recognition is nothing new, and in this paper we will show its convenience for such work.

The rest of paper is structured as follows. In Section Ⅱ, some related works by others are described. The details of the implementation are given in Section Ⅲ. The experimental results and comparisons with previous works are represented in Section Ⅳ. Presented experiments are summarized in Section Ⅴ.

II. Related Work

The MSER has a great application in object recognition, image retrieval, scene classification and so on since J. Matas originally proposed MSER in [2]. In this paper, extremal regions are constructed as 4-neighbourhood regions by an extremal property of the intensity function in the region and on its outer boundary. J. Matas implements car license plate detection by CESR in [3]. Hao Wooi Lim and Yong Haur Tay detect license plate character in nature scene with MSER and SIFT unigram classifier in [4]. Jong Bae Kim implements a license recognition system based on MSER and machine learning approach – SVM, in [5]. And from many previous works described in [6], it can be seen that character recognition with neural networks is already a relatively mature field.

III. Approch

The license plates are often shown by white characters on blue background or black on yellow. In order to improve the accuracy rate, we first locate the license plate based on the color characteristics of the plate and perform tilt correction (this won't be necessary if in the next step we can accurately extract characters). Due to the specificity of the method, in this paper, we only focus on the previous condition, but we will still discuss possible methods when it comes to yellow background license plates.

As there is an obvious difference between background and characters of plate in binary image, the plate region is a typical extrema region. With the pixels in characters varying steadily, these regions satisfy the maximally stable extremal condition and they are MSERs. There are other MSERs in the image, so how to eliminate these regions and preserve the plate region are the key steps in the car license plate detection.

According to the results of license plate detection, interference regions have diversity in shape and are, and may be affected by the light condition, therefore these interference regions are uncertain. If the interference regions can be removed as many as possible, the license plate character regions can be obtained.

Finally, we cut out the characters based on MSER areas and normalize them, and feed them into the trained neural network for recognition.

A more detailed solution is described step by step here, and Fig. 3 shows a flow chart of the whole process.

A. License Plate Location Detection

Unlike some common methods like edge detection, this algorithm directly looks for connected regions in the image where the color, shape and texture match the license plate features. By analyzing a large number of license plate images, it can be found that for pixels with certain target colors, they can be filtered out directly by setting a range for the H, S and V components, without the need for more complicated color distance calculation, which can save a lot of time in color segmentation.

The experimental data shows the HSV range of the blue license plate: Hue between 200 and 255 degrees, saturation between 0.4 and 1, value between 0.3 and 1.

Since the Euclidean distance between two points in RGB triple primary color space is not linearly proportional to the color distance, the positioning range in the blue area cannot be well controlled when setting. Thus, the recognition rate decreases when there are more blue backgrounds in the picture. In this paper, an adaptive adjustment scheme is applied. The segmented area is adjusted for recognition. The candidate regions are located several times according to the aspect ratio and blue-white ratio. Finally, the license plate area is found.

B. *Tilt Correction*

The license plate image acquired after license plate positioning will inevitably have some degree of skew, which will not only bring difficulties to the next step of character segmentation, but will also have a direct impact on the correct rate of license plate recognition eventually. Here, we use the Radon transform method, which is the projection of the image intensity along a radial line oriented at a specific angle.

MATLAB provides native support in the Image Processing Toolbox for Radon transform [7]. R = radon(I) returns the Radon transform R of 2-D grayscale image I for angles in the range [0, 179] degrees.

C. *MSER Detection*

Maximum intensity region is a region that all the value of pixels of the boundary are less than that belonging to it, and minimum intensity region is greater than that. From a sequence of nested extremal regions, extremal region is maximally stable if and only if $q(i) = |Q_{i+\triangle} - Q_{i-\triangle}|/|Q_i|$, the maximally stable extremal condition. Such regions are of interest since they possess the following properties: invariance to affine transformation of image intensities, covariance to adjacency preserving, stability, multi-scale detection and the set of all extremal regions can be enumerated in $O(n \log\log n)$, where n is the number of pixels in the image.

Also, MATLAB provides support for the MSER features detection in the Computer Vision Toolbox [8]. The MSER detector incrementally steps through the intensity range of the input image to detect stable regions. The ThresholdDelta parameter determines the number of increments the detector tests for stability. Think of the threshold delta value as the size of a cup to fill a bucket with water. The smaller the cup, the more number of increments it takes to fill up the bucket. The bucket can be thought of as the intensity profile of the region.

(see Fig. 1.)

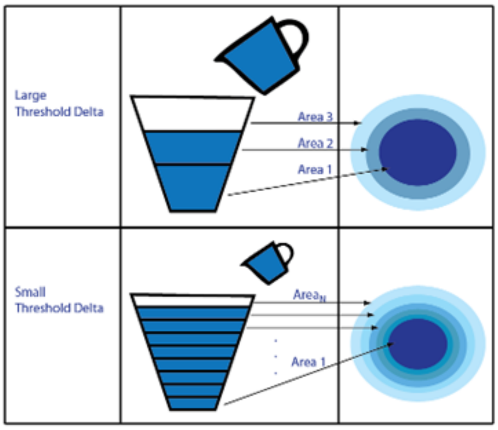

Fig. 1. The bucket analogy of the MSER algorithm

The MSER object checks the variation of the region area size between different intensity thresholds. The variation must be less than the value of the MaxAreaVariation parameter to be considered stable.

At a high level, MSER can be explained, by thinking of the intensity profile of an image representing a series of buckets. Imagine the tops of the buckets flush with the ground, and a hose turned on at one of the buckets. As the water fills into the bucket, it overflows and the next bucket starts filling up. Smaller regions of water join and become bigger bodies of water, and finally the whole area gets filled. As water is filling up into a bucket, it is checked against the MSER stability criterion. Regions appear, grow and merge at different intensity thresholds.

The MATLAB code was developed with reference to the official MathWorks documentation [9], which is about automatically detect and recognize text in natural images.

D. *Eliminate Interference Regions*

Although the MSER algorithm picks out the characters, it also detects many other stable regions in the image that are not characters (Fig. 2).

Fig. 2. The colorful regions indicate the MSER

Here, we set up two filters on MSER, and the principle of the two filters is basically the same: we perform a binary connected components analysis on the input MESR, and then filter out some non-character regions based on the size and aspect ratio of the connected components.

After all the non-character regions are removed, we cut out the characters and normalize them to 32 × 16 binary images for subsequent input to the neural network

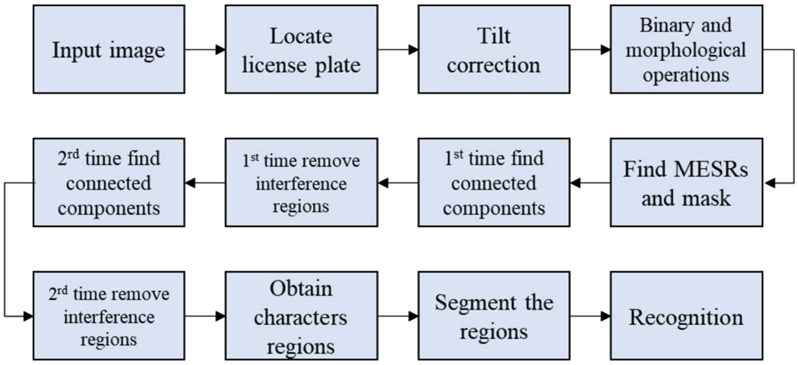

Fig. 3. Flow chart

E. Recognition

If all the previous processes were carried out correctly, we should now have seven 32×16 binary images, one of them is Chinese character and the others are numbers and English characters. The neural network used here refers to Andrew Ng's neural network open class practice problems [10], including one hidden layer, using each pixel value of the binary image for input, and finally using the sigmoid function for output (Fig. 4). Due to the special characteristics of English and Chinese characters, two models are built and trained separately.

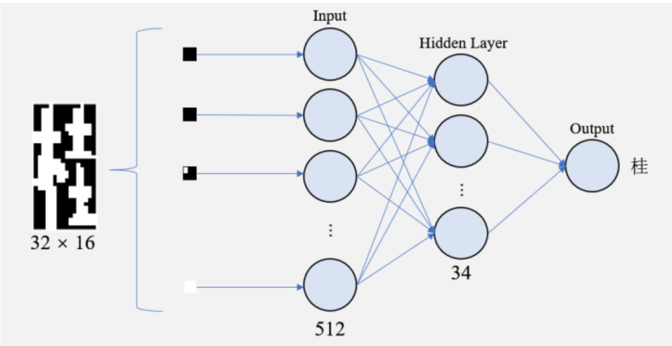

Fig. 4. Neural network diagram

IV. Experiments and Results

The previous section describes the general process. In this section, we will present the results of each step and the details of the implementation. All steps are divided into three steps according to operations: License plate image pre-processing, MSER detection and processing, and characters recognition.

A. License Plate Image Prerocessing

First, we use the method mentioned above to locate the license plate. Due to the special principle of the algorithm, it is decided to recognize the license plate mainly for family compact cars with white characters on blue background. However, because the Euclidean distance between two points in the RGB trichromatic space is not linearly proportional to the color distance, it cannot be well controlled when setting the localization range of the blue area. In this way, the recognition rate decreases in the case of more blue backgrounds in the picture, and the license plate area cannot be extracted effectively.

In this paper, an adaptive adjustment scheme is applied. The segmented area is adjusted for recognition. The candidate regions are located several times according to the aspect ratio and blue-white ratio.

After finding the license plate area, we use the Radon method mentioned before to correct the tilt. This is for the characters segmentation process in the future.

In MATLAB, binary images are often represented as 0 and 1 arrays. We binarize the obtained license plate image to facilitate further processing of the image, making it simple and with a reduced amount of data to highlight the contours of the target of interest.

This is indicated more graphically in Fig. 5.

Fig. 5. Steps of license plate image preprocessing

B. MSER Detection and Processing

According to Fig. 2, there are many interference regions if we directly use the function detectMSERFeatures provided by MATLAB. Therefore we set up two filters, where they are called coarse and fine filtering respectively. We perform a connected components analysis on the obtained MSER region binary image by first coarse filtering some regions that obviously do not match the characters, and then we perform a closed operation on the filtered image. After the closing operation, another fine filtering is performed, and finally the enclosing box that encloses the text region is obtained.

(1) Coarse Filtering

Firstly, we use a mask to take out the number of coordinate coefficients of the MSERs and assign the corresponding coefficients to true. Therefor we obtain the MSERs.

To implement coarse filtering, we build a function which mainly analyzes the input mask for the connected components, then filters out some non-text regions, and finally returns the binary image with the non-text filtered out and the candidate values for the close operation.

We first filter out areas that are too small and too large (more than 30% of the whole frame), and then filter out areas that do not satisfy the aspect ratio (width / height < 0.1 or width / height > 2). Note that here, since the images we get are not completely uniform (e.g. from the same camera at the same distance), the parameters we use do not satisfy every case.

Here we add a judgment. We know that Chinese license plates have 7 characters, if we detect 7 character regions in this step, we will not perform the next region filtering step, on the contrary, if there are more than seven, we will perform the next filtering step.

Algorithm 1 roughly describes this step.

Algorithm 1 MSER Connected Components Analysis and Processing (Coarse Filtering)

Input: MSER mask and binary license plate image

Output: Processed MSER mask and segmented images

```
function CompAnalysis (MSERmask, lpbwimg)
    ConnectComp ← findConnected(MSERmask);
    ConnectComp.Area ← getArea(ConectComp);
    for i from 1 to number_ConnectComp
        if ConnectComp.Area(i) < 10 || ConnectComp.Area(i) > 0.3 * area_lpbwimg)
          || width / height < 0.1 || width / height > 2
            delete ConnectComp.Area(i);
        else
            SegImgs ← Segment(lpbwimg);
        end if
    end for
    return ConnectComp, SegImgs;
end function
if number_SegImgs > 7
    FineFiltering(ConnectComp);
end if
```

(2) Fine Filtering

This step is similar to the previous one. Once again we look for connected components and further eliminate the non-character regions. Here we filter out the areas that clearly do not match, and if it goes well, there are only seven character areas left.

After the character regions are all detected, we segments the regions images and resize them to 32×16 to be used as input to the neural network next.

This detection and processing methodology is illustrated in Fig. 6 The green rectangle is used to indicate MSER.

Fig. 6. Steps of MSER detection and processing

C. *Characters Recognition*

(1) Neural Network Training

The most important part in the neural network approach is the building and training of the neural network model. However, due to time and environmental constraints, we are not able to obtain a large dataset of license plate images, so we use an open source trained model, which has three layers using sigmoid function to output and trained by numbers of 32 × 16 binary characters images, thanks to the work done by CaptYoung [11].

(2) Recognition

Here, we use the parameters of the neural network that has been trained to recognize the characters. We just need to load the model via MATLAB's Code and input the segmented images we obtained before to complete all the steps.

D. *Graphical User Interface Implemented in MATLAB*

We integrated all the operations and designed a GUI (Graphical User Interface) using MATLAB's app designer. The GUI was designed by referring to MATLAB's official documentation [12].

As for the operation, the user only needs to load the image and click "apply" button, he/she then should see the output image and characters recognition result (some unnecessary intermediate product images are omitted in the GUI). See Fig. 7 for more details and operations guide.

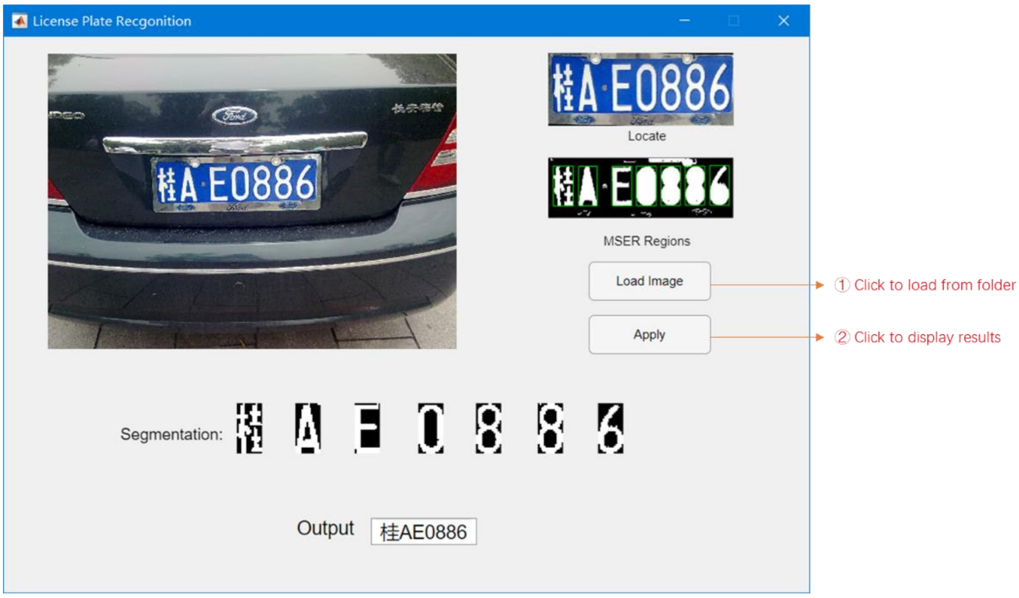

Fig. 7. GUI design

V. Discussion

We will discuss the results in this paper comparing with previous ones from the literature. However, since the work we have done is still very elementary, we will not discuss the mathematical values here, but only compare the advantages and disadvantages of the methods.

In the license location part, there are many other works locating it through its special shape. They use binarization and morphological methods such as and open-close operations to accurately locate license plates. Compared to our method of positioning by color, this method avoids the error that can be caused when there are more blue areas on the screen, and also can be applied to other colors (e.g., yellow), but it can also cause a failure of positioning when there are too many license plate shaped areas on the screen. The relatively better method is to combine the two, and we can get this inspiration from [13].

In this paper, the apply of MSER detection can reduce the work of location [14] (although we did the location work for the accuracy of the results). Compared to the work of some others, such as what mentioned in [1], using a lot of morphological methods to do characters locations, frame removing, characters segmentation etc., our methods eliminates many steps, only need to modify the parameters to get a better result.

As for the recognition part, we believe that the relationship between machine learning and image processing will become more and more inseparable as arithmetic power develops, and from the results in this paper, we can conclude that the machine learning approach is simple and efficient. Although the training of neural networks takes time and computing power compared with the template matching approach, machine learning is certainly becoming a mainstream in academia.

VI. Conclusion

The license plate character regions are the typical MSERs. The combination of MSER methods and machine learning methods proposed in this paper is simple and the result is accurate.

Through this project, we learned and consolidated some knowledge about edge finding, geometric correction, character segmentation, character recognition, based on the content of the image processing class, and reproduced and innovated on the basis of the existing methods, which benefited me a lot.

Acknowledgment

We thank Prof. Zhang for his dedication to us during this semester.

References

[1] Gupta, Dr. Nipin, D. Tayal, D. P. Gupta, D. Goyal and M. Goyal. “A Review: Recognition of Automatic License Plate in Image Processing.” ,2017.

[2] Matas J, Chum O, Urban Metal, “Robust wide baseline stereo from maximally stable external regions,” Image and Vision Computing, 2004, 22(10): 761-767.

[3] Jiri Matas, Karel Zimmermann. “Unconstrained License Plate and Text Localization and Recognition,” Czech Technical University Prague Center for Machine Perception Proceedings of the 8th International IEEE Conference on Intelligent Transportation Systems Vienna, Austria, September 13-16, 2005.

[4] Hao Wooi Lim , Yong Haur Tay, “Detection of License Plate Characters in Natural Scene with MSER and SIFT unigram Classifier”, 2010.

[5] Kim JB. (2012) MSER and SVM-Based Vehicle License Plate Detection and Recognition System. In: Lee G., Howard D., Ślęzak D., Hong Y.S. (eds) Convergence and Hybrid Information Technology. ICHIT 2012. Communications in Computer and Information Science, vol 310. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-32692-9_66.

[6] Liu Q. et al. (2017) A Review of Image Recognition with Deep Convolutional Neural Network. In: Huang DS., Bevilacqua V., Premaratne P., Gupta P. (eds) Intelligent Computing Theories and Application. ICIC 2017. Lecture Notes in Computer Science, vol 10361. Springer, Cham. https://doi.org/10.1007/978-3-319-63309-1_7.

[7] MathWorks Help Center, Documentaion of Radon transform, https://ww2.mathworks.cn/help/images/ref/radon.html?lang=en

[8] MathWorks Help Center, Documentaion of detectMSERFeatures, https://ww2.mathworks.cn/help/vision/ref/detectmserfeatures.html

[9] MathWorks Help Center, Documentaion of Automatically Detect and Recognize Text in Natural Images, https://ww2.mathworks.cn/help/vision/ug/automatically-detect-and-recognize-text-in-natural-images.html

[10] Coursera, Machine Learning taught by Andrew Ng, provided by Stanford University. https://www.coursera.org/learn/machine-learning

[11] https://github.com/CaptYoung/License-Plate-Recognition/tree/master/Training

[12] MATLAB app desginer. https://www.mathworks.com/products/matlab/app-designer.html

[13] G. Li, R. Yuan, Z. Yang and X. Huang, "A Yellow License Plate Location Method Based on RGB Model of Color Image and Texture of Plate," Second Workshop on Digital Media and its Application in Museum & Heritages (DMAMH 2007), Chongqing, 2007, pp. 42-46, doi: 10.1109/DMAMH.2007.35.

[14] W. Wang, Q. Jiang, X. Zhou and W. Wan, "Car license plate detection based on MSER," 2011 International Conference on Consumer Electronics, Communications and Networks (CECNet), XianNing, 2011, pp. 3973-3976, doi: 10.1109/CECNET.2011.5768335.