

A Novel Least Square and Image Rotation based Method for Solving the Inclination Problem of License Plate in Its Camera Captured Image

ChangCheng Wu*, Hao Zhang, JiaFeng Hua, Sha Hua, YanYi Zhang, XiaoMing Lu, YiChen Tang

¹Traffic Management Research Institute of the Ministry of Public Security
Number 88 QianRong RD, WuXi City, JiangSu Province, China
[e-mail: dingsouqing@163.com]

*Corresponding author: ChangCheng Wu

Received April 8, 2019; accepted June 3, 2019; published December 31, 2019

Abstract

Recognizing license plate from its traffic camera captured images is one of the most important aspects in many traffic management systems. Despite many sophisticated license plate recognition related algorithms available online, license plate recognition is still a hot research issue because license plates in each country all round the world lack of uniform format and their camera captured images are often affected by multiple adverse factors, such as low resolution, poor illumination effects, installation problem etc. A novel method is proposed in this paper to solve the inclination problem of license plates in their camera captured images through four parts: Firstly, special edge pixels of license plate are chosen to represent main information of license plates. Secondly, least square methods are used to compute the inclined angle of license plates. Then, coordinate rotation methods are used to rotate the license plate. At last, bilinear interpolation methods are used to improve the performance of license plate rotation. Several experimental results demonstrated that our proposed method can solve the inclination problem about license plate in visual aspect and can improve the recognition rate when used as the image preprocessing method.

Keywords: inclination, license plate, rotation, least square method

1. Introduction

License plates permit vehicles traveling on a road within the national territory. Each country all round the world lacks of uniform format. Fig. 1 shows an example about license plate in China [1]-[2]. Kinds include private, training, police, embassy, new energy etc. Sizes include 440cm×220cm, 440cm×140cm, 220cm×140cm, 220cm×140cm etc. Background color include yellow, blue, white, black etc. For characters, English letters are A to Z except for I and O, Arabic numbers are 0 to 9, and Chinese character are mainly the province abbreviations.



Fig. 1. Different kinds, sizes and formats about license plates in China

In order to monitor vehicles on road, several countries have bound lof cameras on the road sections and road intersections to capture license plate images in recent years, bringing the development of several related methods. Palaiahnakote Shivakumara et al. [3] combined the convolutional neural networks and recurrent neural networks for license plate classification and recognition. Alexandre Perez et al. [4] combined four well known classifiers and four multiple classifiers to recognize human action. A. Jalal et al. [5] utilized R transformation of depth silhouettes to recognize human daily activities. Md.Taufeeq Uddin et al. [6] proposed random forests to recognize human activities and postural transitions on the smartphone. AhmadJalal et al. [7] proposed multi-fused features to recognize human activities from continuous sequences of depth map. Faisal Farooq et al. [8] proposed a method to recognize human facial expression by feeding hybrid features to self-organizing maps. Shohei Suzuki et al. [9] proposed four directional feature fields as a feature extraction method for character recognition in nail camera system. Ahmad Jalal et al. [10] proposed depth silhouettes to recognize daily activities when recognizing human home activity. Shaharyar Kamal et al. [11] proposed addresses shape and motion features approach to track and recognize human silhouettes. Karpuravalli Srinivas Raghunandan et al. [12] proposed symmetry features based on stroke width to classify license plate. Adnan Farooq et al. [13] proposed a feature structured framework to track and recognize different activities of multiple user. Maxim Karpushin et al.

[14] proposed to use depth information to derive local projective transformations and compute descriptor patches from the texture image.

In summary, all above discussed algorithms have their own characteristics, which provide good results in their respective environments. However, recognizing license plates from their camera captured images is still a hot research issue because license plates in each country all round the world lack of uniform format and their camera captured images are often affected by multiple adverse factors, such as low resolution, poor illumination effects, installation problem etc. In this work, we mainly consider the inclination problem of license plate, namely, license plate is inclined when compared with the horizontal direction, due to the reason of vehicle lane derivation, poor camera installation etc. Considering license plates separated from their camera captured images, our proposed method solve the inclination problem of license plates through four parts: Firstly, special edge pixels of license plates are chosen to represent main information of license plate. Secondly, least square methods are used to compute the inclined angle about license plate [15]-[16]. Then, coordinate rotation methods are used to rotate the license plate [17]-[18]. At last, bilinear interpolation methods are used to improve the performance of license plate rotation. Several experimental results demonstrated that our proposed method can solve the inclination problem of license plate in visual aspect and can improve the recognition rate when used as the image preprocessing method.

This paper is organized as follows. Firstly, least square based method and coordinate rotation based method are described in Section 2. Then, a novel method is proposed in Section 3 to solve the inclination problem of license plate. Section 4 demonstrates the experimental results. Finally, the conclusion is described in Section 5.

2. Overviews of least square method and coordinate rotation method

2.1 Least square method

As shown in Fig. 2, let $X-Y$ denote a rectangular coordinate system and $(x_k, y_k) : k=1, 2, \dots, m-1, m$ denote coordinates of m points arbitrarily distributed on both sides of a straight line $y=ax+b$, where a and b are the line slope and intercept respectively. Least square method (LSM), a straight line fitting method, is mainly used to obtain parameters a and b of $y=ax+b$ by outputting the minimum sum of error square between y_k and (ax_k+b) through the following steps:

1. Defining the objective function $S(a, b)$:

$$S(a, b) = \sum_{k=1}^m [(ax_k + b) - y_k]^2 \quad (1)$$

2. Deriving $S(a, b)$ to obtain line parameters a and b respectively:

$$\begin{cases} \frac{\partial S(a, b)}{\partial (a)} = \frac{\partial \sum_{k=1}^m [(ax_k + b) - y_k]^2}{\partial (a)} = 0 \\ \frac{\partial S(a, b)}{\partial (b)} = \frac{\partial \sum_{k=1}^m [(ax_k + b) - y_k]^2}{\partial (b)} = 0 \end{cases} \Rightarrow \begin{cases} a \sum_{k=1}^m x_k + mb = \sum_{k=1}^m y_k \\ \sum_{k=1}^m ax_k^2 + \sum_{k=1}^m bx_k = \sum_{k=1}^m x_k y_k \end{cases} \Rightarrow \begin{bmatrix} \sum_{k=1}^m x_k & m \\ \sum_{k=1}^m x_k^2 & \sum_{k=1}^m x_k \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \sum_{k=1}^m y_k \\ \sum_{k=1}^m x_k y_k \end{bmatrix} \quad (2)$$

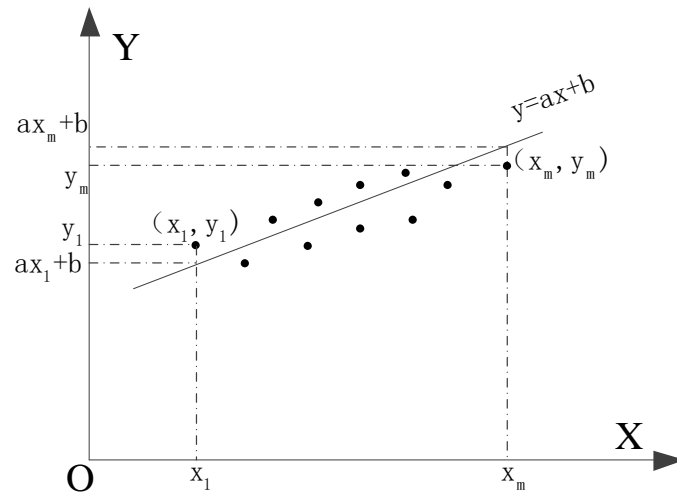


Fig. 2. Points distributed on both sides of a straight line in the rectangular coordinate system

2.2 Coordinate rotation algorithm

As shown in **Fig. 3**, let \vec{OA} denote an vector in X - Y , r denote the space distance between the origin point O and an arbitrarily point A , and ϕ denote the angle between \vec{OA} and X axis. Coordinates of A are $x_A = r \cos \phi$ and $y_A = r \sin \phi$. Let \vec{OB} be another vector which is obtained by rotating \vec{OA} around O of θ angle in counter clockwise direction. Coordinates of point B can be described in Eq.(3).

$$\begin{cases} x_B = r \cos(\phi + \theta) = r \cos \phi \cos \theta - r \sin \phi \sin \theta = x_A \cos \theta - y_A \sin \theta \\ y_B = r \sin(\phi + \theta) = r \cos \phi \sin \theta + r \sin \phi \cos \theta = x_A \sin \theta + y_A \cos \theta \end{cases} \Rightarrow \begin{bmatrix} x_B \\ y_B \end{bmatrix} = \begin{bmatrix} x_A & y_A \end{bmatrix} \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \quad (3)$$

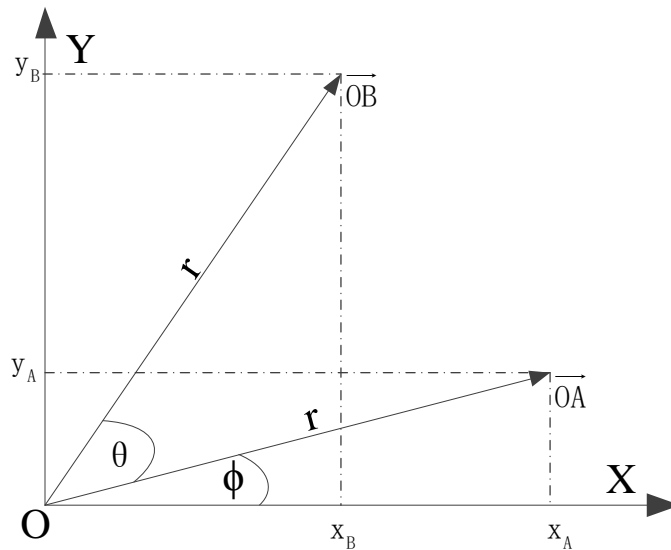


Fig. 3. Coordinate rotation method in a rectangular coordinate system

3. Proposed method

In this paper, since our proposed method highlights algorithm of optimizing inclination problem of license plates in their camera captured images, their segmentation and recognition methods are not the focus of this paper, but can be consulted in Ref.[19], Ref.[20] and Ref.[21]. Our proposed method consisted of four parts as shown in Fig. 4:

Part1, called “License plate edge finding operation,” finds the edge of license plate through two steps: transforming the input license plate image from RGB space to gray space, and finding the edge of license plate according to its gray image in step1.

Part2, called “License plate angle computation operation,” computes the inclined angle of license plate through two steps: finding the rectangle information area of license plate to choose some special edge pixels as samples for computing its inclined angel, and using LMS method to compute the inclined angle of license plate basing on the coordinates of chosen special edge pixels in Step1.

Part3, called “License plate rotating operation,” rotates license plate based on the inclined angle in Part2 through two steps: using coordinate rotation method to rotate coordinates of the input image basing on the inclined angle in Part2, and replacing pixels of the rotated coordinates with pixels of the coordinates before rotating as the output license plate image.

Part4, called “License plate reconstruction operation,” reconstructs the bad performance in Part3 through two steps: using coordinate rotation method to rotate coordinates of some not well rotated pixels in Part3, and using the bilinear interpolation method to further improve the performance of output image.

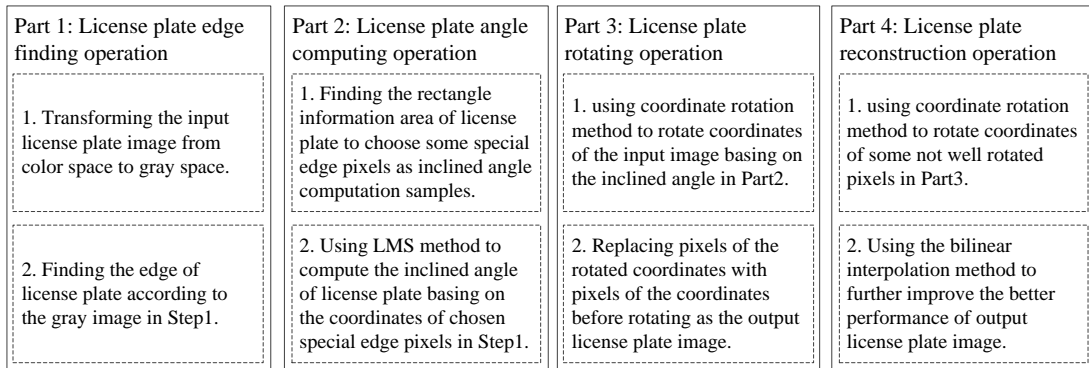


Fig. 4. Structure diagram of our proposed method

3.1 Edge information finding operation

Fig. 5 shows the definition about inclined angel of license plate. Dotted line 1 and dotted line 2 denote the horizontal direction and the upper edge of license plate respectively. Compared with the horizontal direction, the intersected angle θ between dotted line 1 and dotted line 2 denotes the inclined angle of license plate.



Fig. 5. Definition about the inclined angle of license plate

Let U and U^p denote images of license plate in color and grey space respectively. At each location (i, j) , transforms between color pixel value $U_{i,j}$ and gray pixel value $U^p_{i,j}$ can be described in Eq.(4).

$$U^p_{i,j} = U_{i,j}^R \times 0.299 + U_{i,j}^G \times 0.587 + U_{i,j}^B \times 0.114 \quad (4)$$

where $U_{i,j}^R, U_{i,j}^G, U_{i,j}^B$ are the RGB components of $U_{i,j}$. Let U^b denote an edge image of U^p with each pixel value described in Eq.(5).

$$U^b_{i,j} = \begin{cases} 255 & |U^p_{i+1,j} - U^p_{i,j}| \geq \text{threshold} \\ 0 & |U^p_{i+1,j} - U^p_{i,j}| < \text{threshold} \end{cases} \quad (5)$$

where $U^b_{i,j}$ denotes result about judging whether $U^p_{i,j}$ is an edge pixel with threshold.

If the gradient between $U^p_{i+1,j}$ and $U^p_{i,j}$ is larger than threshold, $U^p_{i,j}$ can be judged as an edge pixel of license plate. **Fig. 6a** and **Fig. 6b** are the images of license plate in color and gray space respectively. **Fig. 6c** is the gradient image of **Fig. 6b**. **Fig. 6d, Fig. 6e** and **Fig. 6f** are the edge images of **Fig. 6b** with threshold=20, threshold=50 and threshold=100 respectively. Seen from **Fig. 6d** to **Fig. 6f**, when threshold increases from 20 to 100, the chosen edge pixels become gradually disappeared.

Eq.(5) is mainly used to find the edge pixels which will be computed as the samples of Eq.(8) to compute the inclined angle of license plate. However, if threshold is set too small, more edge pixels like in **Fig. 6d** will be chosen as the samples of Eq.(8), representing more edge information but costing large computation. Conversely, if threshold is set too large, less edge pixels like in **Fig. 6f** will be chosen as the samples of Eq.(8), reducing computation cost but representing little edge information. Therefore, threshold cannot be set too large or too small. Like **Fig. 6e**, we usually adopts threshold $\in [40, 60]$ as parameters of Eq.(5) to obtain the edge pixels, not only representing edge information of license plate but also reducing computation cost in Eq.(8).

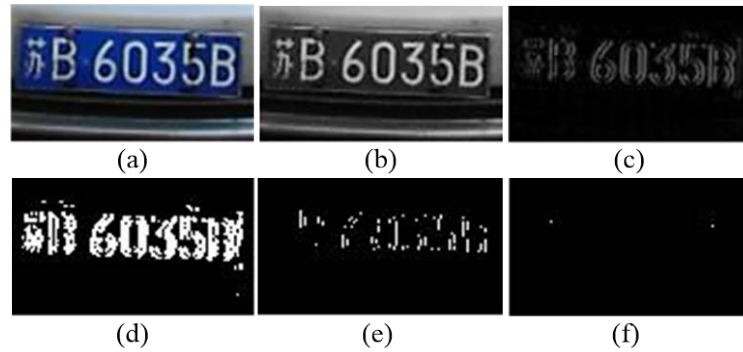


Fig. 6. Example about obtaining edge pixels of license plate:

(a) color image, (b) gray image, (c) gradient image,
(d) edge image (threshold=20), (e) edge image (threshold=50), (f) edge image (threshold=100)

3.2 Inclined angle computing operation

Basing on the edges of license plate, steps of computing θ are as follows:

Step1: finding information area

Usually, as shown in **Fig. 6a**, a license plate image usually include not only license plate itself but also other parts, such as the front bumper, the bonnet etc. Hence, license plate itself area, denoted “rectangle” in U^β , is enough to consider its inclined angle, with “up,” “down,” “left” and “right” being its four sides:

$$\left\{ \begin{array}{ll} \text{up:} & \sum_{t=1}^{H_1} U_{\text{up},t}^\beta \geq 0, \quad 0 \leq \text{up} \leq \frac{1}{3} W \\ \text{down:} & \sum_{t=1}^{H_1} U_{\text{down},t}^\beta \geq 0, \quad \frac{2}{3} W \leq \text{down} \leq W \\ \text{left:} & \sum_{t=1}^{W_1} U_{t,\text{left}}^\beta \geq 0, \quad 0 \leq \text{left} \leq \frac{1}{7} H \\ \text{right:} & \sum_{t=1}^{W_1} U_{t,\text{right}}^\beta \geq 0, \quad \frac{6}{7} H \leq \text{right} \leq H \end{array} \right. \quad (6)$$

Let us take **Fig. 6e** as an example of finding its “rectangle” information area in license plate. When **Fig. 6e** is applied by Eq.(6), “up=15” and “down=43” as shown **Fig. 7a**, “left=8” and “right=108” as shown **Fig. 7b**, together combining the “rectangle” area of **Fig. 7c**. Compared **Fig. 6e** with the “rectangle” area of **Fig. 7c**, image size is decreased from 112×64 to 101×29. Therefore, 4239 pixels are not necessary to be considered as the information of license plate when computing its inclined angle in the following steps.

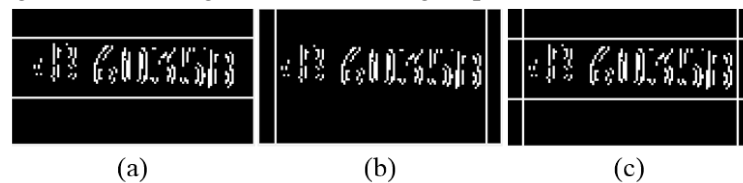


Fig. 7. Example about finding “rectangle” information area:

(a) “up=15” and “down=43”, (b) “left=8” to “right=108”, (c) “rectangle” area

Step2: choosing special edge pixel

Although “rectangle” area contains almost the only information of license plate itself, it still includes more edge pixels and will cost large computations when using Eq.(8) to compute the inclined angle of license plate. Therefore, in order to not only reduce computation cost but also represent the edge information, Eq.(7) is used to choose “special” edge pixels inside “rectangle” area as the samples of Eq.(8) to compute the inclined angle of license plate.

$$\begin{cases} \text{special} & \text{if } U_{i,j}^p = 255, U_{i+1,j}^p = 255, U_{i+2,j}^p = 255, U_{i+3,j}^p = 255 \\ \text{no-special} & \text{otherwise} \end{cases} \quad (7)$$

According to Eq.(7), if pixels, in four continuous rows and one column, are all edge pixels, the one in the first row can be seen as the “special” sample. The red cycled pixel in Fig. 8a is an example of “special” pixel, because other samples, aligned in the row 3 & column 4, row 4 & column 3, row 5 & column 3, are all edge pixels. When Fig. 8b is operated by Eq.(7), the found “special” samples are shown in Fig. 8c. Compare with Fig. 8b which has 351 edge pixels in the “rectangle” area, Fig. 8c only use 36 edge pixels to represent the inclined information of license plate, further reducing computation on 315 edge pixels when using Eq.(8) to compute the inclined angle of license plate.

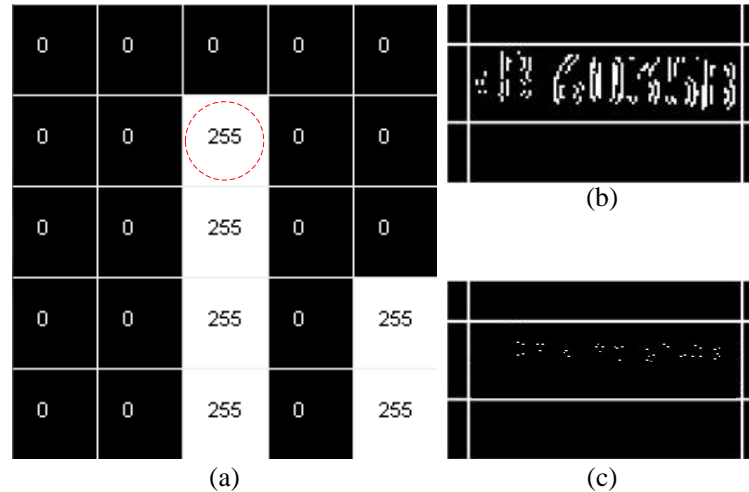


Fig. 8. Example about “special” edge pixel: (a) an enlarged part of Fig. 8b, (b) “rectangle” area, (c) “special” edge pixel

Step3: computing the inclined angle

According to the coordinates of “special” edge pixels, θ can be computed as following by Eq.(2):

$$\theta = a \times 180 / \pi = \left(\frac{\sum_{k=1}^{tol} k(i) \times k(j) - \frac{1}{tol} \sum_{k=1}^{tol} k(i) \sum_{k=1}^{tol} k(j)}{\sum_{k=1}^{tol} k(i)^2 - \frac{1}{tol} (\sum_{k=1}^{tol} k(i))^2} \right) \times 180 / \pi \quad (8)$$

where tol denote number of “special” edge pixel, $k(i)$ and $k(j)$ denote the horizontal coordinate and the vertical coordinate of k th “special” edge pixel respectively. When pixel coordinates of Fig. 7c are applied by Eq.(8), its slope angle is -0.0207 in circular space, i.e. -1.19° in angle space. It means that Fig. 7c is inclined -1.19° when compared with the horizontal direction.

3.3 License plate rotating operation

License plate rotation, operation of rotating the input image U around its center $(0.5W_1, 0.5H_1)$ by θ angle, included Part A and Part B as following:

Part A: coordinate rotating map

This part mainly rotates the coordinates of U to that of output image OUT around the center of U by θ angel through the following steps.

1. Mapping coordinate (i, j) in $I-J$ to coordinate (x, y) in $X-Y$.

$$[x, y, 1] = [i, j, 1] \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ -0.5W_1 & 0.5H_1 & 1 \end{bmatrix} \quad (9)$$

2. Rotating one point (x_1, y_1) around $(x=0, y=0)$ by θ angle to the other point (x_2, y_2) in counter clockwise direction according to Eq.(3) in $X-Y$.

$$[x_2, y_2, 1] = [x_1, y_1, 1] \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (10)$$

3. Mapping coordinate (x, y) in $X-Y$ to coordinate (\hat{i}, \hat{j}) in $\hat{I}-\hat{J}$.

$$[\hat{i}, \hat{j}, 1] = [x, y, 1] \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0.5W_2 & 0.5H_2 & 1 \end{bmatrix} \quad (11)$$

where H_1 and W_1 are the height and width of input image U , H_2 and W_2 are the height and width of output image OUT , (i, j) and $I-J$ denote the pixel coordinate and coordinate system of U respectively, (\hat{i}, \hat{j}) and $\hat{I}-\hat{J}$ denote the pixel coordinate and coordinate system of OUT respectively. Since coordinate rotation is based on rotating license plate around its center pixel, center coordinate of $I-J$ is the origin coordinate of $X-Y$, i.e. $(i=0.5W_1, j=0.5H_1)$ of $I-J$ is $(x=0, y=0)$ of $X-Y$. From step1 to step3, coordinates rotating mapping, from the input image to the output image, can be described as following:

$$[\hat{i}, \hat{j}, 1] = [i, j, 1] \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ -0.5W_1 & 0.5H_1 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0.5W_2 & 0.5H_2 & 1 \end{bmatrix} \quad (12)$$

Fig. 8 demonstrates an coordinate rotation example from U to OUT . **Fig. 8a** shows a 5×5 pixels image with $W_1=4$ and $H_1=4$. According to Eq.(9), $x=i-2$ and $y=2-j$. **Fig. 8b** shows that $(i, j) \neq (2, 2)$ is $(x, y) \neq (0, 0)$. **Fig. 8c** shows that $(x_1, y_1) \neq (1, 1)$ changes to $(x_2, y_2) \neq (0, \sqrt{2})$ when rotated around $(0, 0)$ by $\theta=45^\circ$ in counter clockwise direction. $W_2=4\sqrt{2}$ and $H_2=4\sqrt{2}$. **Fig. 8d** shows that $\hat{i}=x_2 + 0.5W_2 = 2\sqrt{2}$ and

$\hat{j} = y_2 + 0.5H_2 = \sqrt{2}$. That is to say that $(i, j) = (2, 2)$ in the input image is mapped into $(\hat{i}, \hat{j}) = (2\sqrt{2}, \sqrt{2})$ in the output image.

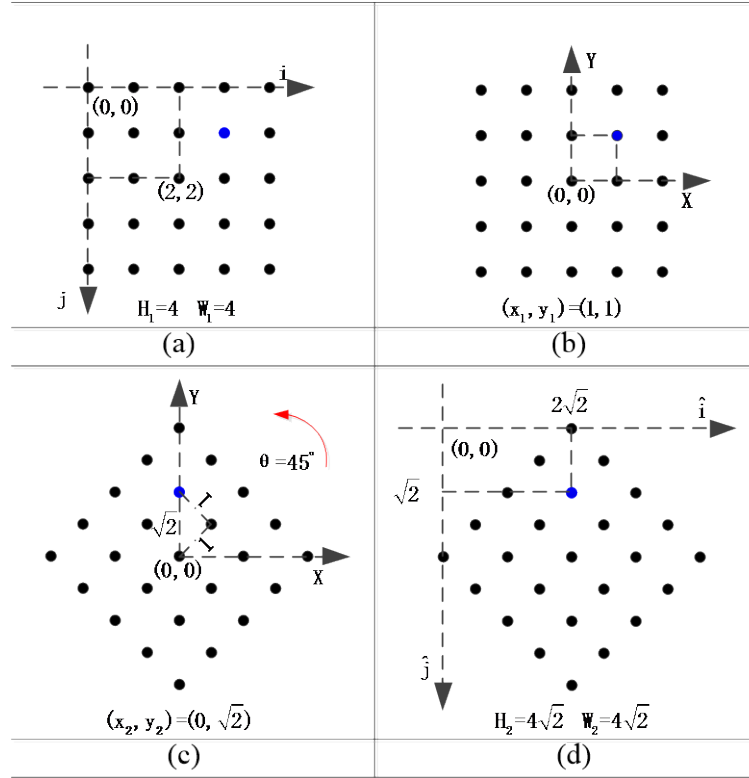


Fig. 8. Coordinate rotation example from \mathbf{U} to \mathbf{WT} :

- (a) input image coordinate system $\mathbf{I}-\mathbf{J}$, (b) mathematical coordinate system $\mathbf{X}-\mathbf{Y}$,
(c) coordinate rotating in counter clockwise direction, (d) output image coordinate system $\hat{\mathbf{I}}-\hat{\mathbf{J}}$

Part B: pixel replacing

Basing on the coordinate rotation map in Part A, each output pixel value $\mathbf{WT}_{\hat{i}, \hat{j}}$ is defined as follows:

$$\mathbf{WT}_{\hat{i}, \hat{j}} = \mathbf{U}_{i, j} \quad (13)$$

If (\hat{i}, \hat{j}) is an integer number coordinate, $\mathbf{WT}_{\hat{i}, \hat{j}}$ is replaced by $\mathbf{U}_{i, j}$. However, if (\hat{i}, \hat{j}) is an irrational coordinate, $\mathbf{WT}_{\hat{i}, \hat{j}}$ cannot be replaced by $\mathbf{U}_{i, j}$ because (\hat{i}, \hat{j}) is not a real image coordinate. Fig. 9a is the obtained “rectangle” area of Fig. 6a. Fig. 9b is the result about rotating Fig. 9a around its center pixel by its inclined angle -1.19° . Seen from Fig. 9b, the red marked pixels are the samples whose coordinate map is from the integer number in \mathbf{U} to the irrational number in \mathbf{WT} .



Fig. 9. Example about license plate rotating result:
(a) “rectangle” area, (b) image rotation

3.4 License plate reconstructing operation

License plate reconstruction, operation of reconstructing the red marked pixels like in **Fig. 9b**, included Part A and Part B as follows:

Part A: coordinate rotating

This part mainly rotates the coordinates of output image \mathbf{OUT} to that of input image \mathbf{U} around the center of \mathbf{OUT} by θ angel through the following steps.

1. Mapping coordinate (\hat{i}, \hat{j}) in $\hat{I}-\hat{J}$ to coordinate (x, y) in $X-Y$.

$$[x, y, 1] = [\hat{i}, \hat{j}, 1] \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ -0.5W_2 & 0.5H_2 & 1 \end{bmatrix} \quad (14)$$

2. Rotating one point (x_1, y_1) around $(x=0, y=0)$ by θ angle to the other point (x_2, y_2) in clockwise direction according to Eq.(3) in $X-Y$.

$$[x_2, y_2, 1] = [x_1, y_1, 1] \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (15)$$

3. Mapping coordinate (x, y) in $X-Y$ to coordinate (i, j) in $I-J$.

$$[i, j, 1] = [x, y, 1] \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0.5W_1 & 0.5H_1 & 1 \end{bmatrix} \quad (16)$$

Similar to Eq.(12), coordinate mapping, from the output image to the input image, can be described as following:

$$[i, j, 1] = [\hat{i}, \hat{j}, 1] \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ -0.5W_2 & 0.5H_2 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0.5W_1 & 0.5H_1 & 1 \end{bmatrix} \quad (17)$$

Compared with Section 3.2, Step1 and Step3 are the same, while the rotation directions in Step2 are opposite. **Fig. 10** demonstrates an example about coordinate rotation from \mathbf{OUT} to \mathbf{U} . **Fig. 10a** is the same as **Fig. 7d** with $W_2=4\sqrt{2}$ and $H_2=4\sqrt{2}$. According to Eq.(9), $x=\hat{i}-2\sqrt{2}$ and $y=2\sqrt{2}-\hat{j}$. **Fig. 10b** shows that $(\hat{i}, \hat{j})=(2\sqrt{2}, 2\sqrt{2})$ is $(x, y)=(0, 0)$. **Fig. 10c** shows that $(x_1, y_1)=(0, \sqrt{2})$ changes to $(x_2, y_2)=(1, 1)$ when rotated around $(0, 0)$ by

$\theta = 45^\circ$ in clockwise direction. $W_1=4$ and $H_1=4$. Fig. 10d shows that $\hat{i} = x + 0.5W_1 = 3$ and $\hat{j} = y + 0.5H_1 = 1$. That is to say that $(\hat{i}, \hat{j}) = (2\sqrt{2}, 2\sqrt{2})$ in the input image is mapped into $(i, j) = (3, 1)$ in the output image.

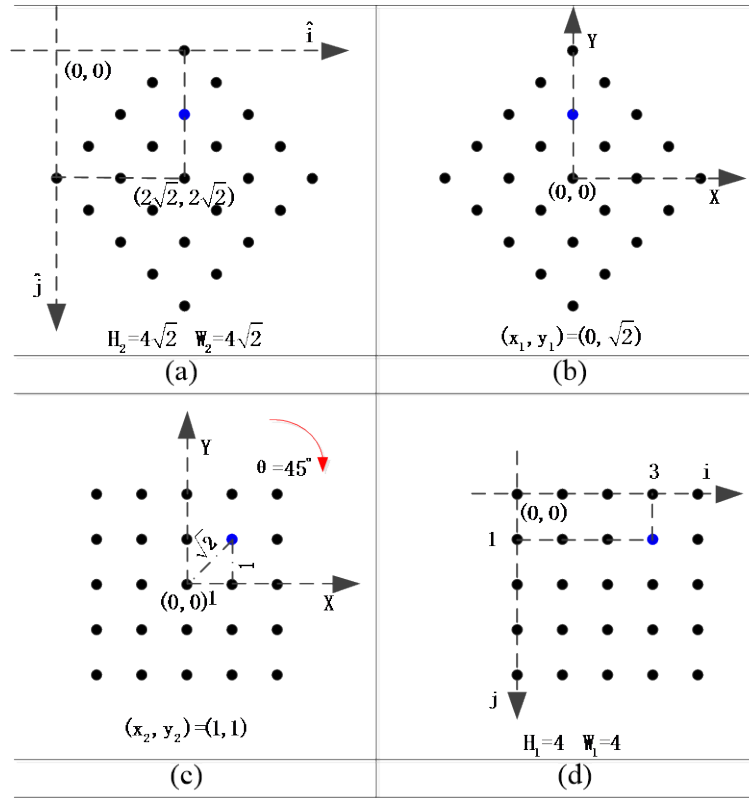


Fig. 10. Coordinate rotation example from \mathcal{U} to $\hat{\mathcal{U}}$:

- (a) output image coordinate system $\hat{\mathbf{I}} - \hat{\mathbf{J}}$, (b) mathematical coordinate system $\mathbf{X} - \mathbf{Y}$,
(c) coordinate rotating in clockwise direction, (d) input image coordinate system $\mathbf{I} - \mathbf{J}$.

Part B: pixel evaluating

For the pixels which are not well rotated like the red marked ones in Fig. 9b, basing on the coordinate mapping from (\hat{i}, \hat{j}) to (i, j) , we make the neighbors of $\mathcal{U}_{i,j}$ with bilinear interpolation operation to evaluate their outputs as follows:

$$\mathcal{U}_{i,j} = d_1 \times d_2 \times \mathcal{U}_{i,j}^1 + d_1 \times (1-d_2) \times \mathcal{U}_{i,j}^2 + (1-d_1) \times d_2 \times \mathcal{U}_{i,j}^3 + (1-d_1) \times (1-d_2) \times \mathcal{U}_{i,j}^4 \quad (18)$$

where $\mathcal{U}_{i,j}^1, \mathcal{U}_{i,j}^2, \mathcal{U}_{i,j}^3$ and $\mathcal{U}_{i,j}^4$ are the upper & left, the lower & left, the upper & right, and the lower & right neighbors about $\mathcal{U}_{i,j}$, d_1 and d_2 are space distance between $\mathcal{U}_{i,j}$ and $\mathcal{U}_{i,j}^1$. If (i, j) is an irrational number, coordinates of its four neighbors are all integer numbers. For example, if $(i, j) = (2\sqrt{2}, \sqrt{2})$, $\mathcal{U}_{i,j}^1 = \mathcal{U}_{2,1}$, $\mathcal{U}_{i,j}^2 = \mathcal{U}_{2,2}$, $\mathcal{U}_{i,j}^3 = \mathcal{U}_{3,1}$ and $\mathcal{U}_{i,j}^4 = \mathcal{U}_{3,2}$.

Fig. 11 is an example about license plate reconstruction result. Firstly, Fig. 11a, the same as Fig. 9b, is rotated 1.19° around the center pixel in clockwise direction. Then, two pixels,

marked red in Fig. 11a, are evaluated by samples in Fig. 9a according to Eq.(18), outputting the reconstruction performance as shown in Fig. 11b.



Fig. 11. Example about license plate reconstruction result:
(a) image rotation, (b) image reconstruction

From all the above, each step of our algorithm can be described as follows:

1. Finding the edge of license plate basing on its gray image U^p .
2. Using the coordinates of special pixels in license plate as the samples of LMS to compute the inclined angle θ .
3. Rotating (i, j) of U around $(i=0.5W, j=0.5H)$ by θ angle in counter clockwise direction to (\hat{i}, \hat{j}) of OUT and outputting each pixel value $OUT_{\hat{i}, \hat{j}} = U_{i, j}$.
4. Rotating (\hat{i}, \hat{j}) of OUT around $(\hat{i}=0.5W, \hat{j}=0.5H)$ by θ angle in clockwise direction to (i, j) of U on pixels which are not well rotated in Step3 and evaluating them by bilinear interpolation method according to Eq.(18).

4. Experiment Result and Discussion

4.1 Experimental environment

For evaluating the performance of our proposed method, we take our experiment on a real road under different weather and different light condition. Fig. 12 shows a description of the environment. The pole is installed near the road intersection, and the cameras are placed upon the pole 6 meters above the ground. On each road lane, when vehicles are about 20 to 30 meters far away from the projection of the pole, vehicles will be captured by the cameras. Usually, speeds about the moving vehicles are around 80~100km/h. In addition, our proposed algorithm is installed in cameras by C++ language under the embedded Linux system.

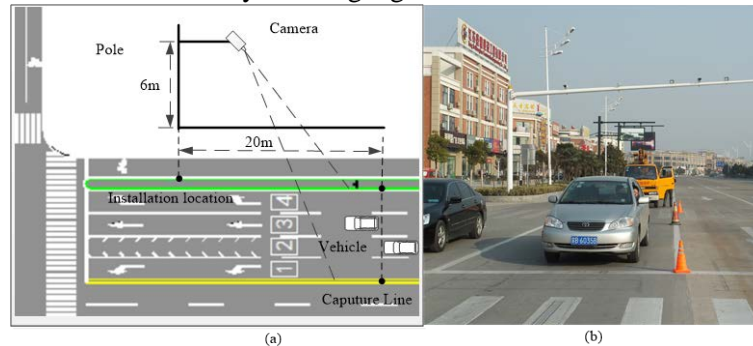


Fig. 12. Experimental framework: (a) sketch map, (b) real environment

4.2 Different kinds vehicle experimental

In order to show the visual performance of our proposed method, [Fig. 13](#), [Fig. 14](#) and [Fig. 15](#) show the results of our proposed method on the kinds of the private, the coach, the training and the police vehicles. In each figure, images in the first row are input license plates which are separated from their camera captured images, images in the second row are edges of input images, images in the third row are rectangle character areas of license plates, images in the fourth row are results of our proposed methods. Seen from the fourth row images in each figure, our proposed method can solve the inclination problem of license plates in better visual performances by separating license plate accurately and solving their inclination problems.



Fig. 13. Experiment about our proposed method on the private vehicles



Fig. 14. Experiment about our proposed method on the coach vehicles



Fig. 15. Experiment about our proposed method on the training and the police vehicles.

4.3 Different impact experiment

In addition, **Fig. 16** shows the performance of our proposed method when the experiment environment is affected by negative factors. The driver face is marked red to protect its privacy in the whole experiment. **Fig. 16a** and **Fig. 16d** demonstrate the experiment under the condition when vehicles deviate from road lane, with license plates inclined in their traffic camera images. **Fig. 16b**, **Fig. 16e**, **Fig. 16c** and **Fig. 16f** demonstrate the experiment under the condition of rain and night which may influence the brightness, color and definition of license plates. Seen from the bottom of each picture, the left images are license plates which are separated from their camera captured images, the right images are the results of our proposed method by separating license plates from the left images accurately and solving their inclination problems. That is to say that the propose method can overcome the bad influence of lane deviation and adverse illumination effects etc.



Fig. 16. Experiment about our proposed method on different negative factors: (a) and (b) lane deviation, (c) and (e) rain condition, (d) and (f) night condition.

4.4 Recognition performance analysis

To examine the performance of our proposed method in improving license plate recognition rate, we design the experiment by using or not using our proposed method before implementing some latest methods to recognize license plate. License plate kinds include the private, coach, training and police, and quantity of each kind license plate throughout the whole experiment is 1000. Seen from **Table 1**, if “NO” using our proposed algorithm to solve the inclination problem of license plates before implementing Ref.3 to recognize license plates, for license plate recognition performance on the private, the coach, the training and the police license plates respective, mistakes on English letters are 32, 29, 36, 45, and mistakes on Arabic

numbers are 30, 28, 37, 42. Conversely, if “YES” using our proposed method to solve the inclination problem of license plates before implementing Ref.3 to recognize license plates, mistakes on English letters decrease to 25, 21, 24, 23, and mistakes on Arabic numbers decrease to 21, 18, 21, 18. Suppose each wrong recognized license plate has only one mistake in English letter or Arabic number, license plate recognition rate of Ref.3 increased from 93.0% when “NO” using our method to 95.7% when “YES” using our method. When the left methods in **Table 1** are the same analyzed like Ref.3, comparing “NO” using our proposed method with “YES” using our proposed method, license plate recognition rates of Ref.22 and Ref.23 are increased from 92.7% to 95.9%, and from 93.1% to 95.8% respectively. Therefore, we can say that our proposed method can increase the license plates recognition rate when used as the image preprocessing method.

Table 1. Performance of “No” or “YES” using our proposed method before recognizing license plate

Methods	Whether using our algorithm	English Letters Mistake				Arabic Numbers Mistake			
		Private	Coach	Training	Police	Private	Coach	Training	Police
Ref.[3]	NO	32	29	36	45	30	28	37	42
	YES	25	21	24	23	21	18	21	18
Ref.[22]	NO	39	35	38	43	32	29	36	40
	YES	23	23	21	20	25	16	17	18
Ref.[23]	NO	29	36	40	35	36	34	31	36
	YES	21	19	19	21	16	23	23	25

5. Conclusion

In this work, a novel way is proposed to solve the inclination of the license plate in their camera captured images. Firstly, special edge pixels are chosen as the samples of least square method to compute the inclination angle of license plates. Then, the license plate is rotated according to its computed inclination angle and around its center pixels. At last, bilinear interpolation methods are used to improve the performance of license plate rotation. Several experimental results demonstrated that our proposed method can solve the inclination problem about license plate in visual aspect and can improve the recognition rate when used as the image preprocessing method.

Acknowledge

Thanks the reviewers for giving good advice to help us improve the quality of this paper. The work is supported by the “Public Security Theory and Soft Science Research Program (2018LLYJGAJT043)” project.

Reference

- [1] License plates of motor vehicles of the People's Republic of China. *GA/36*.
- [2] Christos-Nikolaos E Anagnostopoulos, Ioannis E. Anagnostopoulos, Ioannis D. Psoroulas, Vassili Loumos, and Eleftherios Kayafas, "License Plate Recognition From Still Images and Video Sequences: A Survey," *IEEE Trans. On Intelligent Transportation Systems*, Vol.9, no. 3, pp.377-391, 2008. [Article \(CrossRef Link\)](#)
- [3] Palaiahnakote Shivakumara, Dongqi Tang, Maryam Asadzadehkaljahi, Tong Lu, Umapada Pal, Mohammad Hossein Anisi, "CNN-RNN based method for license plate recognition," *CAAI Transactions on Intelligence Technology*, Vol.3, No.3, pp.169-175, July, 2018. [Article \(CrossRef Link\)](#)
- [4] Alexandre Perez, Hedi Tabia, David Declercq, Alain Zanotti, "Feature covariance for human action recognition," in *Proc. of International Conference on Image Processing Theory, Tools and Applications*., December 2016. [Article \(CrossRef Link\)](#)
- [5] A. Jalal, Md. Zia Uddin, T.-S. Kim, "Depth video based human activity recognition system using translation and scaling in variant features for life logging at smart home," *IEEE Transactions on Consumer Electronics*, Vol.58, No.3, pp.863 –871, August 2012. [Article \(CrossRef Link\)](#)
- [6] Md.Taufeeq Uddin, Md.Mutlaleb Billah, Md.Faisal Hossain. "Random forests based recognition of human activities and postural transitions on smart phone," in *Proc. of International Conference on Informatics, Electronics and Vision*, May 2016. [Article \(CrossRef Link\)](#)
- [7] AhmadJalal, Yeon-HoKim, Yong-JoongKim, ShaharyarKamal, DaijinKim. "Robust human activity recognition from depth video using spatiotemporal multi-fused features," *Pattern Recognition*, Vol.61, pp.295-308, 2017. [Article \(CrossRef Link\)](#)
- [8] Faisal Farooq, Jalal Ahmed, Lihong Zheng, "Facial expression recognition using hybrid features and self-organizing maps," in *Proc. of IEEE International Conference on Multimedia and Expo*., July 2017. [Article \(CrossRef Link\)](#)
- [9] Shohei Suzuki, Kunihito Kato, Kazuhiko Yamamoto, "A consideration of effective feature extraction method for Nail camera system," in *Proc. of 17th Korea-Japan Joint Workshop on Frontiers of Computer Vision*., March 2011. [Article \(CrossRef Link\)](#)
- [10] Ahmad Jalal, Md. Zia Uddin, Jeong Tai Kim, Tae-Seong Kim, "Recognition of human home activities via depth silhouettes and R transformation for smart homes," *Indoor and Built Environment*, Vol. 21, No.1, pp. 184-190, September 2011. [Article \(CrossRef Link\)](#)
- [11] Ahmad Jalal, Shaharyar Kamal, Daijin Kim, "Shape and Motion Features Approach for Activity Tracking and Recognition from Kinect Video Camera" in *Proc. of 29th International Conference on Advanced Information Networking and Applications Workshops*, pp.445-450, March 2015. [Article \(CrossRef Link\)](#)
- [12] Karpuravalli Srinivas Raghunandan, Palaiahnakote Shivakumara, Lolika Padmanabhan, Govindaraju Hemantha Kumar, Tong Lu, Umapada Pal, "Symmetry features for license plate classification," *CAAI Transactions on Intelligence Technology*, Vol.3, No.3, pp.176 –183, November 2018. [Article \(CrossRef Link\)](#)
- [13] Adnan Farooq, Ahmad Jalal, Shaharyar Kamal, "Dense RGB-D Map-Based Human Tracking and Activity Recognition using Skin Joints Features and Self-Organizing Map," *KSII Transactions on internet and information system*, Vol.9, No.5, pp.1856-1869, May 2015. [Article \(CrossRef Link\)](#)
- [14] Maxim Karpushin, Giuseppe Valenzise, Frédéric Dufaux. "Local visual features extraction from texture depth content based on depth image analysis," in *Proc.of IEEE International Conference on Image Processing*, October 2014. [Article \(CrossRef Link\)](#)
- [15] P. Vanicek, "Further Development and Properties of the spectral Analysis by Least squares," *Astrophysics & Space Science*, vol.12, no.1, pp. 10-33, 1971. [Article \(CrossRef Link\)](#)
- [16] N. R. Lomb, "Least squares frequency analysis unequally spaced data," *Astrophysics and space science*, vol. 39, pp. 447-462, 1976. [Article \(CrossRef Link\)](#)
- [17] A. Acharyya, K. Maharatna, B. M. Al-Hashimi, J. Reeve, "Coordinate Rotation Based Low Complexity N-D FastICA Algorithm and Architecture," *IEEE Transactions on. Signal Process.*, vol. 59, no. 8, pp. 3997-4011, 2011. [Article \(CrossRef Link\)](#)

- [18] Lie-Chung Shen, Jyh-Ching Juang, Ching-Lang Tsai, Ching-Liang Tseng, "Monitoring Water Levels and Currents Using Reflected GPS Carrier Doppler Measurements and Coordinate Rotation Model," *IEEE Transactions on Instrumentation and Measurement*, vol.59, no.1, pp.153-163, 2010. [Article \(CrossRef Link\)](#)
- [19] S. G. Kim, H. G. Jeon, H. I. Koo, "Deep learning based license plate detection method using vehicle region extraction," *Electronics Letters*, vol.53, pp. 1034-1036, 2017. [Article \(CrossRef Link\)](#)
- [20] Yule Yuan, Wenbin Zou, Yong Zhao, Xinan Wang, Xuefeng Hu, Nikos Komodakis, "A robust and efficient approach to license plate detection," *IEEE Transactions on Image Processing*, vol.26, pp. 1102-1114, 2017. [Article \(CrossRef Link\)](#)
- [21] Bo Li, Bin Tian, Ye Li, Ding Wen, "Component based license plate detection using conditional random field model," *IEEE Transactions on Intelligent Transportation Systems*, vol.14, pp. 1690-1699, 2013. [Article \(CrossRef Link\)](#)
- [22] Xiang Bai, Cong Yao, WenYu Liu, "Strokelets: a learned multi-scale mid-level representation for scene text recognition," *IEEE Transactions on Image Processing*, vol.25, no.6, pp. 2789-2802, 2016. [Article \(CrossRef Link\)](#)
- [23] Yun Yang, Donghai Li, Zongtao Duan, "Chinese vehicle license plate recognition using kernel-based extreme learning machine with deep convolutional features," *IET Intelligent Transport Systems*, Vol.12, No.3, pp.213-219, 2018. [Article \(CrossRef Link\)](#)



ChangCheng Wu received the B.S. degree from Dalian Railway Institute, Dalian City, Liaoning Province, China, in 2003, the M.S. degree from Dalian JiaoTong University, Dalian City, Liaoning Province, China, in 2006, and the Ph.D. degree from Shanghai JiaoTong University, Shanghai, China, in 2010. He is currently a research associate in Traffic Management Research Institute of the Ministry of Public Security. His current research interests include traffic technology monitoring, image processing and artificial intelligence.



Hao Zhang received his Engineering M.S. degree from Southeast University, Nanjing, China, in 2008. He is presently a Director of the Detection and Research Department at Traffic Management Research Institute of the Ministry of Public Security. His current research interests include traffic monitoring technology, information security technology and software testing technology.



Jia Feng Hua received his B.S. degree from SanJiang University, NanJing, China in 2009. He is now a engineer at the Traffic Management Research Institute of the Ministry of Public Security. His intrests included road traffic safety product testing technology and standardization.



Sha Hua received her B.S degree from Nanjing University of Technology, NanJing, China in 2014. And in 2017 she received her M.S degree from Nanjing University of Aeronautics and Astronautics, China. She is now a software testing engineer at the Traffic Management Research Institute of the Ministry of Public Security. Her research interests include software testing, image processing and machine learning.



Yanyi Zhang received the B.S. degree from Southeast University, Nanjing, China, in 2011, and the M.S. degree from Worcester Polytechnic Institute, Massachusetts, USA, in 2013. His current research interests include image processing, pattern recognition and artificial intelligence.



Xiaoming Lu received his M.S. degree from Suzhou University in 2012. His research interest includes Image Processing, pattern recognition, and artificial intelligence.



Yichen Tang received his B.S. and M.S. degrees in Instrument Science and Engineering from Southeast University, Nanjing, China, in 2012 and 2015. He is currently an engineer at Inspection Department of TMRI, Wuxi, China. His current research interests include image processing, pattern recognition, and computer vision.