

A Two-Stage Vehicle Type Recognition Method Combining the Most Effective Gabor Features

Wei Sun^{1,2,*}, Xiaorui Zhang^{2,3}, Xiaozheng He⁴, Yan Jin¹ and Xu Zhang³

Abstract: Vehicle type recognition (VTR) is an important research topic due to its significance in intelligent transportation systems. However, recognizing vehicle type on the real-world images is challenging due to the illumination change, partial occlusion under real traffic environment. These difficulties limit the performance of current state-of-art methods, which are typically based on single-stage classification without considering feature availability. To address such difficulties, this paper proposes a two-stage vehicle type recognition method combining the most effective Gabor features. The first stage leverages edge features to classify vehicles by size into big or small via a similarity k-nearest neighbor classifier (SKNNC). Further the more specific vehicle type such as bus, truck, sedan or van is recognized by the second stage classification, which leverages the most effective Gabor features extracted by a set of Gabor wavelet kernels on the partitioned key patches via a kernel sparse representation-based classifier (KSRC). A verification and correction step based on minimum residual analysis is proposed to enhance the reliability of the VTR. To improve VTR efficiency, the most effective Gabor features are selected through gray relational analysis that leverages the correlation between Gabor feature image and the original image. Experimental results demonstrate that the proposed method not only improves the accuracy of VTR but also enhances the recognition robustness to illumination change and partial occlusion.

Keywords: Vehicle type recognition, improved Canny algorithm, Gabor filter, k-nearest neighbor classification, grey relational analysis, kernel sparse representation, two-stage classification.

1 Introduction

Automatic recognition of vehicle type plays an important role in enabling vehicle specific

¹ School of Automation, Nanjing University of Information Science & Technology, Nanjing, 210044, China.

² Jiangsu Collaborative Innovation Center on Atmospheric Environment and Equipment Technology, Nanjing University of Information Science & Technology, Nanjing, 210044, China.

³ Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science & Technology, Nanjing, 210044, China.

⁴ Rensselaer Polytechnic Institute, Troy, NY 12180, USA.

* Corresponding Author: Wei Sun. Email: sunw0125@163.com.

Received: 26 June 2020; Accepted: 22 July 2020.

operations in intelligent transportation systems, including applications of traffic surveillance, electronic toll collection, and intelligent parking. Various studies conducted in the last decade sought to improve the performance of vehicle type recognition (VTR) [Soon, Khaw, Chuah et al. (2019); Liu, Wang, Gong et al. (2019)]. The major determinants of the performance of VTR relate to methods for extracting vehicle features and the design of vehicle type classifier, which are the focuses of research on VTR.

For effective feature extraction, many methods have been applied to VTR based on the advancements in computer vision. For instance, edge-based methods [Lin, Chan, Fu et al. (2012); Fabrizia and Renata (2012)] can extract the global geometrical contour of an image accurately and quickly. Applying the edge features to VTR yields a fast VTR method due to the low computational complexity. However, these methods are sensitive to illumination and image background variations, which can significantly reduce the accuracy of VTR in a complex travel environment with low illumination. As a result, the extracted features are limited to preliminary recognition in VTR.

Facing these challenges, several feature extraction methods apply mathematical transformation to enhance the adaptability of VTR in a complex travel environment. For example, the histogram of oriented gradients (HOG) descriptor [Wang and Cai (2016)] calculates the gradient and magnitude for every pixel in a vehicle image and combines these values into a histogram of oriented gradients according to their weights. Features extracted by the scale-invariant feature transform (SIFT) [Dong and Jia (2013); Ambardekar, Nicolescu and Bebis (2014)] and speeded up robust features (SURF) descriptors [Chen, Hsieh, Yan et al. (2015)] are adopted in VTR, due to their robustness to the changes of scale, translation, and rotation of vehicle images. The Gabor filter [Adi and Arnida (2015)] is another commonly adopted descriptor to extract local features of vehicle image. Compared to edge-based methods, the feature extraction method based on Gabor filter can extract structural details of vehicle image from multiple scales and orientations. More importantly, the Gabor filter is insensitive to illumination change or scale variation. The multi-resolution and multi-orientation properties, as well as the insensitivity to travel environment, make the Gabor filter a power solution for VTR to resist external disturbance such as illumination change, partial occlusion or scale variation.

Although these methods based on mathematical transformation can extract structural details, they also generate a large amount of information irrelevant to vehicle classification, which increases computation burden and degrades the reliability of classifier. A potential solution is to select the most effective features to reduce the dimension and redundancy of the extracted local features.

In addition to the feature extraction and selection, classifier design is the other determinant affecting the performance of VTR. Typical VTR classifiers in the literature apply the k -nearest neighbors (kNNs) [Fabrizia and Renata (2012); Gu and Lee (2012)], artificial neural network (ANN) [Chen, Gong, Xie et al. (2017)], support vector machine (SVM) [He, Sang, Gao et al. (2017); Kachach and Maria (2016); Fang, Zhou, Yu et al. (2017)], hidden Markov model (HMM) [Zhou, Deng and Lv (2011)], and sparse representation classifier (SRC) [Gao, Ma and Yuille (2017)].

In classifier design, the kNNs classifier applies a simple classification principle and does not need training in advance. When the number of training samples increases, its

computational cost increases significantly. Therefore, the kNNs classifier is limited to preliminary classification based on simple features such as edge or color. The methods based on ANN or SVM can utilize global features or local features in VTR. However, training these classifiers requires a large number of samples with different types of vehicles; while the trained parameters are easy to fall into local optima that adversely affect the classification accuracy and performance of VTR. The HMM classifier has a good performance in VTR by training HMM mode and optimizing mode parameters. However, the additional computation for optimizing these mode parameters impairs its real-time performance. Compared to other classifiers, the SRC has been widely used in face recognition and classification due to its advantages in not involving complex parameter training and using original image samples as a dictionary without additional transformation. One challenge of the SRC is that some samples are linearly inseparable in the original feature space. The kernel sparse representation-based classifier (KSRC) [Gao, Ma and Yuille (2017); Li, Feng, Chen, et al. (2017); Li and Zhu (2018)] is a nonlinear extension of the SRC [Wang, Shen, Li et al. (2018); Li, Tang, Peeta et al. (2019)]. It can convert linearly inseparable samples in the original feature space to be linearly separable in the new feature space by multiplying a kernel function [Li, Liu, Wang et al. (2019)]. The linear separability improves the accuracy and reliability of the SRC.

Although the aforementioned classifiers can be applied to recognize different types of vehicles, they do not factor the effectiveness of the extracted vehicle features, which can introduce unnecessary computations and reduce the efficiency and reliability of VTR in real-time applications. In addition, all these classifiers adopt a single-stage scheme that classifies vehicle type based on all types of vehicle samples. When the number of vehicle types to be recognized increases, the single-stage methods require a huge amount of training samples, which inevitably increase the difficulty of classifier design.

To address the aforementioned limitations, we propose a VTR method that integrates the most effective Gabor features into a two-stage classification scheme, where the most effective Gabor features are selected as local features based on grey relation analysis (GRA). In the two-stage scheme, edge features are obtained by the improved Canny edge detection algorithm with discrete cosine transformation (DCT). Preliminary vehicle classification is implemented based on the edge features and similarity k-nearest neighbor classifier (SKNNC), and further precise classification is realized based on the most effective Gabor features and the KSRC. More specifically, the first stage applies the SKNNC with cosine similarity calculation to classify vehicles by size into large or small classes, leading to an accurate and reliable preliminary classification. Based on the most effective Gabor features, the second stage introduces the KRSC to classify large-size vehicles into bus or truck and small-size vehicles into sedan or van. The second stage also incorporates a verification and correction step using the minimum residual analysis to enhance the accuracy and robustness of VTR when the vehicle is partially occluded.

In summary, the main contributions of this study are twofold. First, a GRA based method is proposed to select the most effective Gabor features, which reduces the dimension of the feature vector and eliminates potential data redundancy. Second, our proposed two-stage classification scheme leverages two types of features and classifiers to perform VTR, which simplifies the classifier design and enhances the accuracy and robustness of VTR.

The rest of this paper is organized as follows. The next section presents the global and local feature extraction methods as well as the feature selection of local features based on GRA. Section 3 describes the two-stage classification scheme for VTR. Experiments and analyses are shown in Section 4 to illustrate the efficacy of the proposed VTR method. The final section summarizes this study and potential future research directions.

2 Feature extraction

2.1 Extraction of edge features

The edge of vehicle image usually contains rich contour information. In this paper, we extract the edge features in the preliminary VTR.

Canny operator is the most widely used operator in edge detection. However, the captured vehicle images are easily corrupted by high-frequency noises such as salt and pepper noise, Rayleigh noise and Gaussian noise, in the real world traffic. The traditional Canny edge detection method based on Gaussian filtering function is limited for the images corrupted by Gaussian noise [Li, Tang, Peeta et al. (2019); Li, Chen, Peeta et al. (2020)]. When there exist other noises, the method may fail to detect some real edges. Therefore, we propose to replace the Gaussian filtering step in the traditional Canny algorithm with a new correction strategy of discrete cosine transform coefficient to accommodate other types of noises. The key steps are as follows.

1) *DCT transformation*. Assume the size of the original vehicle image $I(x, y)$ is $M \times N$, the DCT transformation can be expressed as:

$$F(u, v) = c(u)c(v) \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} I(x, y) \cdot \cos \left[\frac{(x+0.5)\pi}{M} u \right] \cos \left[\frac{(y+0.5)\pi}{N} v \right] \quad (1)$$

$$c(u) = \begin{cases} \sqrt{1/M}, & u = 0 \\ \sqrt{2/M}, & u \neq 0 \end{cases} \quad (2)$$

$$c(v) = \begin{cases} \sqrt{1/N}, & v = 0 \\ \sqrt{2/N}, & v \neq 0 \end{cases} \quad (3)$$

where, $u, x = 0, 1, 2, \dots, M-1$; $v, y = 0, 1, 2, \dots, N-1$.

2) *DCT coefficient correction*. After the DCT based on Eq. (1), the low-frequency components contain useful information of vehicle image, while the high-frequency components contain noises. A correction strategy for the DCT coefficients is applied to remove high-frequency noises and remain useful information, as formulated by Eq. (4).

$$F'(u, v) = \frac{F^3(u, v)}{F^2(u, v) + \zeta} \quad (4)$$

where ζ is a correction coefficient. Because the DCT coefficient $F(u, v)$ of noise is large, the revised DCT coefficient $F'(u, v)$ becomes even larger. On the contrary, the

revised DCT coefficient $F'(u, v)$ of the useful information becomes smaller when it has a small DCT coefficient $F(u, v)$. Based on this observation, we can remove the noises by setting an appropriate threshold defined by Eq. (5):

$$F''(u, v) = \begin{cases} F'(u, v), & \text{if } F'(u, v) \leq T_C \\ 0, & \text{if } F'(u, v) > T_C \end{cases} \quad (5)$$

In this paper, we let $\zeta = 60$ and $T_C = 1150$ to obtain a good denoising result.

3) *Inverse discrete cosine Transformation (IDCT)*. We implement the IDCT for the DCT coefficients to obtain a smoothed image $I'(x, y)$.

$$I'(x, y) = c(u)c(v) \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F''(u, v) \cdot \cos\left[\frac{(x+0.5)\pi}{M}u\right] \cos\left[\frac{(y+0.5)\pi}{N}v\right] \quad (6)$$

Through the above three steps, we can get a smoothed gray image. Then, we can obtain a continuous and complete edge image by the same steps as the traditional Canny edge detection algorithm, such as the image gradient calculation, non-maxima suppression, and edge connection. Finally, we transform the edge image $I'(x, y)$ into a one-dimensional vector y_E by concatenating its columns and define the y_E as the edge feature of the vehicle.

2.2 Extraction of gabor features

The edge feature can be used to classify the vehicle into a big or small vehicle preliminarily. However, to have a more specific vehicle classification, such as sedan, van, bus, or truck, we need to extract other features to represent the local structural details.

2.2.1 Image partition based on key parts

Note that, not all elements of a vehicle face image are valuable for VTR. Only some key elements are available. Additionally, some elements of a vehicle are easily occluded by other vehicles. If we divide a vehicle image into several key patches, we can recognize the vehicle type through other key elements in other non-occluded patches, even when partial occlusion occurs. In this paper, we divide the vehicle image into four non-overlapping patches shown in Fig. 1, where (a) is the original vehicle image, (b) is the vehicle roof patch, (c) is the windshield and rearview mirror patch, (d) is the hood patch, and (e) is the license plate patch.

2.2.2 Extraction of gabor feature

Gabor wavelets have strong characteristics of spatial locality and orientation. In this paper, the Gabor wavelet is introduced to extract local features in every partitioned patch, which can not only obtain better structural details with multiple scales and multiple orientations but also improve the robustness to illumination changes or partial occlusion [Kuang, Zhang, Jin et al. (2015)]. The Gabor wavelet kernels can be defined:

$$\psi_{u,v}(z) = \frac{\|k\|^2}{\sigma^2} \exp\left[-\frac{\|k\|^2 \|z\|^2}{2\sigma^2}\right] \cdot \left[\exp(ik \cdot z) - \exp\left(-\frac{\sigma^2}{2}\right)\right] \quad (7)$$

where u and v define the orientation and scale of the Gabor kernels, respectively, $z = (x, y)$, $\|\cdot\|$ denotes the norm operator, (x, y) represents the pixel coordinates, and the wave vector $k = \begin{pmatrix} k_v \cos \varphi_u \\ k_v \sin \varphi_u \end{pmatrix}$, with $k_v = k_{\max} / f^v$, $\varphi_u = u \cdot \pi / 8$, k_{\max} representing the maximum frequency, and f denotes the spacing factor between kernels in the frequency domain. In particular, we let $\sigma = 2\pi$, $k_{\max} = \pi / 2$, $f = \sqrt{2}$.

The Gabor features $M_{u,v}(z)$ of a vehicle image can be obtained by convolving the image $I(z)$ with a set of Gabor wavelet kernels defined by Eq. (7) at every pixel (x, y) :

$$M_{u,v}(z) = I(z) * \psi_{u,v}(z) \quad (8)$$

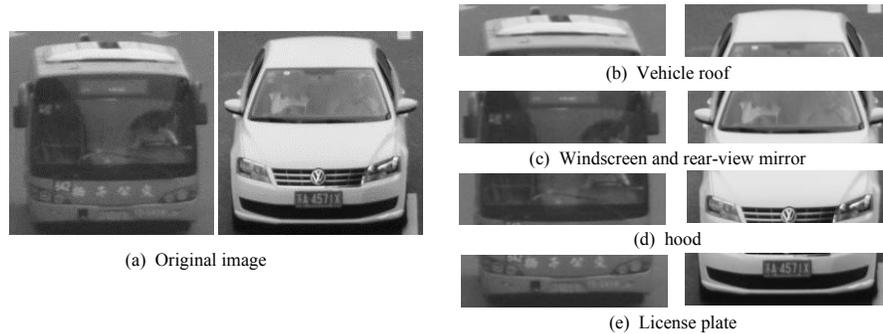


Figure 1: Vehicle image partition

where $I(z)$ expresses the gray image of the vehicle image.

It is common to set the 40 Gabor wavelet kernels with five different scales: $v \in \{0, 1, \dots, 4\}$, and eight orientations: $u \in \{0, 1, \dots, 7\}$. Therefore, we obtain total 40 Gabor feature images based on the 40 Gabor wavelet kernels.

2.2.3 Feature selection based on GRA

These Gabor features of the vehicle extracted from five scales and eight orientations are not suitable for a real-world application due to the large number of dimensions. Additionally, these extracted Gabor features contain redundant information irrelevant to the original vehicle image besides some useful information. The critical question is how to choose the most effective Gabor features to enhance the accuracy and efficiency of VTR. The GRA method is effective in determining the critical elements that significantly influence certain defined objectives due to these advantages such as no restriction of functional form, no requirement for independence or normal distribution, and low computational load. Hence, it has been extensively used for relevance analysis in various disciplines. Here, the GRA is used to perform correlation analysis between the Gabor

feature image and the original image and to select most effective Gabor features. The following steps summarize the details of the algorithm:

1) Assume that $I^j(x, y)$ represents the j th local patch of vehicle. Calculate the amplitude $G_{u,v}^j(x, y)$ as Gabor feature image according to Eq. (7) and Eq. (8).

2) Transform the j th local patch $I^j(x, y)$, $j \in [1, 2, 3, 4]$ and the Gabor features $G_{u,v}^j(x, y)$ extracted from the j th local patch $I^j(x, y)$ into a vector $Y_0^j(k)$ and a vector $Y_l^j(k)$ by concatenating its columns, respectively. Here, $Y_0^j(k)$ represents the original image vector of the k th sample, $k = 1, 2, 3, \dots, r$, r represents the number of image samples, $Y_l^j(k)$ represents the l th Gabor feature vector of the j th local patch of the k th sample, and $l = u \times v$.

3) Grey relation coefficient $\lambda_{0,l}(k)$ between $Y_0^j(k)$ and $Y_l^j(k)$ is calculated as:

$$\lambda_{0,l}(k) = \left| \frac{(\Delta_{\min} + \xi \Delta_{\max})}{(\Delta_{0,l}(k) + \xi \Delta_{\max})} \right| \tag{9}$$

where $\Delta_{0,l}(k) = \|Y_0^j(k) - Y_l^j(k)\|_1$, $\|\cdot\|_1$ represents the 1-norm of vector, Δ_{\min} and Δ_{\max} denote the minimum and maximum values among the $\Delta_{0,l}(k)$, respectively, ξ is resolution coefficient, and $\xi = 0.5$ in this paper.

4) The grey relational degree $\gamma_{0,l}$ between $Y_0^j(k)$ and $Y_l^j(k)$ is calculated as:

$$\gamma_{0,l} = \frac{1}{r} \sum_{k=1}^r \lambda_{0,l}(k) \tag{10}$$

5) Sort the grey relational degree $\gamma_{0,l}$ in and decreasing order to generate a sequence of gray relational degree $\gamma(i)$, where $\{\gamma(1) > \gamma(2), \dots, > \gamma(40)\}$. Then select the first M Gabor feature vectors corresponding with the largest M grey relational degree values and connect these Gabor feature vectors in series to generate the most effective Gabor feature y_G^j in the j th local patch, where $K < 40$. Finally, generate the most effective Gabor feature y_G of the whole vehicle image by connecting the most effective Gabor feature y_G^j of every local patch in series.

3 Vehicle type recognition

3.1 Two-stage classification scheme

Unlike single-stage classification methods that require more training samples and computation time to train classifier parameters, we propose a two-stage classification scheme based on two different types of classifiers and features. In the first stage, the test vehicle is classified into either a big or small vehicle using SKNNC based on the extracted global features. Based on the first stage, the second stage classifies a big vehicle

into either bus or truck and a small vehicle into van or sedan by exploiting the KSRC based on the local features.

Also, we propose to add a verification and correction step based on minimum residual analysis to enhance the reliability of VTR, where a threshold is set to judge whether the classification result in the second stage is reasonable. If the result is reasonable, directly output the classification result in the second stage. Otherwise, we need to change the sub-dataset of large or small vehicles used in the second stage of classification and reclassify the vehicle type based on the have changed sub-dataset and the KSRC. The detailed process of the two-stage classification is summarized into Fig. 2.

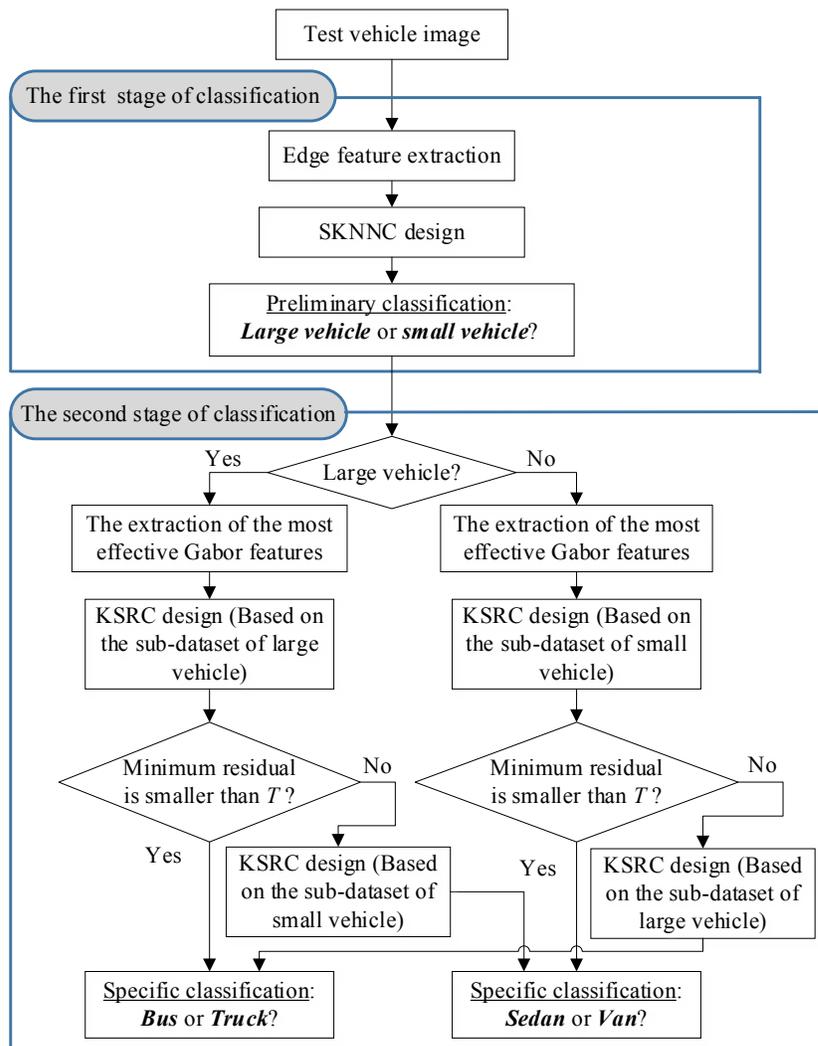


Figure 2: Flowchart of vehicle type recognition

3.2 First stage of classification

The first stage is important in the proposed two-stage classification scheme. It directly determines how well the second stage classification performs. In order to enhance the first stage classification, we propose to combine Euclidean distance and cosine similarity to measure the similarity between the test sample and every training sample during the KNN based classification. Due to considering the similarity of samples in both position and direction, the proposed similarity measure well adapts to the changes of samples such as position or direction. Based on the extracted edge feature and the improved KNN classifier with cosine similarity measure, we can classify the test sample into big vehicle or small vehicle. The steps of the first stage classification are as follows:

1) Assume that the number of vehicle types is M . For each vehicle type there exists n^j ($j=1,2,\dots,M$) samples. Suppose that $a(x,y)$ represents the edge feature extracted from the test sample, and $b^i(x,y)$ indicates the edge feature extracted from the i th training sample in sample set $B = \{b^1(x,y), b^2(x,y), \dots, b^N(x,y)\}$, where $N = \sum_{j=1}^M n^j$.

Referring to Eq. (11), we can calculate the Euclidean distance $D_E(i)$ between $a(x,y)$ and $b^i(x,y)$, for all $i \in \{1,2,\dots,N\}$.

$$D_E(i) = \|a(x,y) - b^i(x,y)\|_2 \quad (11)$$

2) The cosine similarity $S_C(i)$ between the test sample $a(x,y)$ and every training sample $b^i(x,y)$ can be calculated as:

$$S_C(i) = \frac{a(x,y) \times b^i(x,y)}{\sqrt{a(x,y)^2} \times \sqrt{b^i(x,y)^2}} = \frac{a(x,y) \times b^i(x,y)}{\|a(x,y)\| \times \|b^i(x,y)\|} \quad (12)$$

3) The improved similarity measure $S_1(i)$ is defined as:

$$S_1(i) = \frac{S_C(i)}{D_E(i)} \quad (13)$$

Eq. (13) shows that the greater the similarity between the test sample and the i th training sample the higher the value of $S_1(i)$.

4) Find the largest K values of the improved similarity measure $S_1(i)$ and their corresponding vehicle type, i.e., big vehicle or small vehicle, and count the number of every vehicle type corresponding to the largest K values of the similarity measure. Finally, the vehicle type with the largest voting number is deemed as the final classification result.

3.3 Second stage of classification

After the first stage classification, the test sample is classified preliminarily into big vehicle or small vehicle. However, to achieve a more specific vehicle type classification,

a kernel sparse representation based classifier is exploited to classify a big vehicle into bus or truck and a small vehicle into van or sedan in the second stage classification.

3.3.1 Sparse representation based classification

SRC needs to train the over-complete dictionary based on the target features, then reconstruct the samples using the trained dictionary and recognize the target according to the minimum residual between the target and the reconstructed samples. Suppose $A = [A_1, A_2, \dots, A_K] \in \mathfrak{R}^{m \times n}$ is the set of training samples, where A_i is the subset of training samples from class i , $A_i = [S_{i,1}, S_{i,2}, \dots, S_{i,n_i}] \in \mathfrak{R}^{m \times n_i}$, and $S_{i,j}$ is the descriptor of j th sample for the i th class. Given a test sample $y \in \mathfrak{R}^m$, the problem of the sparse representation of y in terms of A can be written as follow:

$$\min_{\beta} \frac{1}{2} \|A\beta - y\|_2^2 + \lambda \|\beta\|_1 \quad (14)$$

where $\|\beta\|_1 = \sum_{j=1}^n |\beta_j|$ is the l_1 norm of β , and $\lambda \geq 0$ is a constant parameter for sparsity.

We can identify y once the solution to Eq. (14) is obtained. The class identity of y is based on the minimization of residuals:

$$\min_c \text{Residual}_c(y) = \|A\delta_c(\beta) - y\|_2^2 \quad (15)$$

where $\delta_c(\beta)$ is a function that selects the coefficient corresponding to the c th class and makes the rest equal to zero. For K classes, there are K functions of $\delta_c(\beta)$, which generate K residuals. Thus SRC algorithm labels y to the class that has the minimum residual.

3.3.2 Kernel sparse representation based classification

Kernel sparse representation [Chen, Wang, Xia et al. (2019); Lin, Feng, Chen et al. (2017); Yang, Shrestha, Li et al. (2018)] is a nonlinear extension of basic sparse representation. Through kernel trick, samples are mapped into a new kernel feature space and then SRC is used in the new feature space. This nonlinear mapping can change the sample distribution. In particular, some samples that are linearly inseparable in the original space can become linearly separable in the new space, which can improve the performance of object classification [Andri, Wang and Tai (2015)]. Therefore, we propose to use the KSRC to classify a big vehicle into bus or truck and classify a small vehicle into van or sedan based on the most effective Gabor features.

Considering the Lasso problem of Eq. (14), we rewrite Eq. (14) as follows:

$$\min_{\beta} \frac{1}{2} \left\| \sum_{i=1}^n \beta_i A_i - y \right\|_2^2 + \lambda \|\beta\|_1 \quad (16)$$

where $\|\beta\|_1$ is the Lasso penalty, and $\lambda \geq 0$ is a constant parameter for sparsity. In this paper, we focus on the Lasso problem of Eq. (16) in kernel space:

$$\min_{\beta} J(\beta) = \frac{1}{2} \left\| \sum_{i=1}^n \beta_i \phi(A_i) - \phi(y) \right\|_2^2 + \lambda \|\beta\|_1 \quad (17)$$

where $\phi(\cdot)$ is an implicit mapping that maps a feature vector to a kernel space. Assume that $\phi(\cdot)$ satisfies $\phi(A)^T \phi(A) = 1$. In order to solve Eq. (17), we introduce a Kernel Coordinate Descent (KCD) algorithm [Jerome, Trevor and Rob (2010)] considering its simplicity and efficiency. Assume that all β_i are fixed except for $i = j$. Minimizing Eq. (17) is equivalent to minimizing Eq. (18):

$$\frac{1}{2} \left\| \beta_j \phi(A_j) - \hat{r}_j \right\|_2^2 + \lambda |\beta_j| \quad (18)$$

where $\hat{r}_j = \phi(y) - \sum_{i=1, i \neq j}^n \hat{\beta}_i \phi(A_i)$ is the partial residual for fitting β_j . Consider $\phi(A_j)^T \phi(A_j) = 1$. Then Eq. (18) is equivalent to:

$$J_j(\beta_j) = \frac{1}{2} \beta_j^2 - \hat{\alpha}_j \beta_j + \lambda |\beta_j| \quad (19)$$

where $\hat{\alpha}_j = \phi(A_j)^T \hat{r}_j$ is the residual correlation. If $\beta_j \neq 0$, taking $\partial J_j(\beta_j) / \partial \beta_j = 0$ in the Eq. (19) can obtain the minimization values when $|\hat{\alpha}_j| > \lambda$, i.e., $\hat{\alpha}_j - \lambda$ for $\beta_j > 0$, and $\hat{\alpha}_j + \lambda$ for $\beta_j < 0$. Otherwise, if $|\hat{\alpha}_j| \leq \lambda$, we have

$$J_j(\beta_j) \geq \frac{1}{2} \beta_j^2 - \hat{\alpha}_j \beta_j + |\hat{\alpha}_j \beta_j| \geq 0 \quad (20)$$

The equality holds if and only if $\beta_j = 0$. Consequently, the coordinate-wise update of β_j is:

$$\beta_j \leftarrow \beta'_j = \begin{cases} \hat{\alpha}_j - \lambda & \text{if } \hat{\alpha}_j > \lambda \\ \hat{\alpha}_j + \lambda & \text{if } \hat{\alpha}_j < -\lambda \\ 0 & \text{if } |\hat{\alpha}_j| \leq \lambda \end{cases} \quad (21)$$

This update, denoted as $\beta_j \leftarrow \text{sign}(\hat{\alpha}_j) (|\hat{\alpha}_j| - \lambda)_+$, is the well-known soft-thresholding shrinkage operation. Note that

$$\hat{\alpha}_j = \phi(A_j)^T \hat{r}_j = \phi(A_j)^T \left[\phi(y) - \sum_{i=1, i \neq j}^n \hat{\beta}_i \phi(A_i) \right] \quad (22)$$

The above equation can be rewritten as:

$$\hat{\alpha} = K(A_j, y) - \sum_{i=1, i \neq j}^n \hat{\beta}_i K(A_j, A_i) \quad (23)$$

where $K(A, y) = \phi(A)^T \phi(y)$ is the kernel function and $K(A, A) = \phi(A)^T \phi(A) = 1$. This is a kernel extension of the covariance update suggested in [28]. Given a kernel function K , we use the KCD algorithm to update β iteratively in the kernel space by applying Eq. (21) and Eq. (23).

For classification, the corresponding kernel SRC criterion as follow:

$$\text{identity} = \arg \min_c \left\| \sum_{l(i)=c} \beta_i \phi(A_i) - \phi(y) \right\|_2^2 = \arg \min_c \delta_c(\beta)^T R \delta_c(\beta) - 2Z^T \delta_c(\beta) \quad (24)$$

where $R = (K(A_i, A_j))_{n \times n}$ is the training kernel matrix, $z = (K(A_i, y))_{n \times 1}$ contains correlation coefficients in the kernel space, and $l(i)$ is the class label of the i th sample. The KSRC algorithm labels y to the class that has the minimum residual.

3.3.3 Vehicle type recognition verification

Because the classification in the second stage relies on the result in the first stage, if the classification in the first stage is false, then the false classification will cause erroneous results in the second stage. Experimental investigation finds that, as long as the employed sub-dataset is correct, the minimum residual in the second stage of classification will not be larger than a given threshold. Namely, in the second stage classification, the minimum residual is larger when the first stage classification is false than the value when the first stage classification is correct. Therefore, we propose to add a verification step to judge whether the classification result in the first stage is accurate or not by setting a threshold T . For a false classification, we re-run the VTR based on the proposed KSRC by changing the employed sub-dataset of large or small vehicle in the second stage classification. The threshold T is determined through a sequence of numerical experiments based on field samples.

The verification step is as follows: (i) if the minimum residual in the second stage is smaller than the set threshold T , then we infer that the classification in the second stage is reasonable and output the classification result directly. (ii) Otherwise, the classification in the second stage is regarded to be unreasonable, and the classification is false in the first stage. For obtaining a correct classification, we change the sub-dataset of large or small vehicle employed in the second stage classification and recognize specific vehicle type based on the changed sub-dataset and the proposed KSRC algorithm.

4 Experiments

To validate the proposed algorithm, we obtained a dataset of 4000 vehicle images containing 1000 bus images, 1000 truck images, 1000 van images, and 1000 sedan images. The proportion of the challenging vehicle images that are partially occluded by other objects or captured in bad condition is about 10% of the whole dataset. The size of

every image is 200×200 pixels. Fig. 3 shows some image examples in the dataset under various conditions. All the experiments were coded and run in Matlab 2014b on a PC with 3GHz i7-2600 CPU and 16GB RAM.

To facilitate the proposed VTR method, all vehicle images were classified into two datasets: big vehicle and small vehicle. The big vehicle dataset is further classified into two sub-sets: bus and truck. The small vehicle dataset is further classified into two sub-sets: van and sedan.

The performance of the VTR is evaluated by the following three typical indicators: precision, recall, and accuracy, defined as follows: $precision = TP / (TP + FP)$, $recall = TP / (TP + FN)$, and $accuracy = (TP + TN) / (TP + FN + FP + TN)$, where, TP, FP, FN and TN are the abbreviations of true positives, false positives, false negatives, and true negatives, respectively, which are defined in [Powers (2011)].

4.1 Selection of the most effective gabor features

We used the Gabor wavelet kernels with five different scales and eight different orientations to extract Gabor features of every partitioned local patch. Fig. 4. shows the extracted Gabor features of the local patch of the license plate.

Then we exploited the proposed GRA algorithm to select the most effective Gabor features. We randomly selected 1000 vehicle images, including four vehicle types from the vehicle image dataset to implement the GRA algorithm. The gray relational degrees between the original partitioned patch images and the Gabor feature images extracted from five scales and eight orientations are shown in Fig. 5. In Fig. 5, the horizontal coordinate indicates the index of the extracted 40 Gabor feature images. The numbers from one to eight in the horizontal coordinate indicate the Gabor feature images when v equals zero and u equals zero to seven, respectively. The numbers from nine to sixteen indicate these Gabor feature images when v equals one and u equals zero to seven, respectively. The numbers from 17 to 24 in the horizontal coordinate indicate these Gabor feature images when v equals two and u equals zero to seven, respectively. The numbers from 25 to 32 in the horizontal coordinate indicate these Gabor feature images when v equals three and u equals zero to seven, respectively. The numbers from 33 to 40 in the horizontal coordinate indicate these Gabor feature images when v equals four and u equals zero to seven, respectively. The last index indicates the mean value of the gray relational degree values of the 40 extracted Gabor features images.

We find that grey relational degree values beyond the mean value mainly concentrate in the ten indexes, such as 17, 23, 25, 31, 32, 33, 34, 35, 39, and 40. It means that the Gabor features extracted from the ten indexes can better describe the vehicle characteristics compared to the Gabor features extracted from other indexes. Therefore, in this paper, we choose the 10 Gabor features corresponding to the ten indexes as the most effective Gabor features (MEGFs). After the MEGFs were selected, the dimension of the Gabor feature of every patch reduced by 75%, from $40 \times 200 \times 50 = 400000$ to $10 \times 200 \times 50 = 100000$.



(a) Fine day



(b) Rainy day



(c) Occluded partial



(d) Dusk and Night

Figure 3: Vehicle images under various conditions

To verify the efficacy of the MEGFs selected by the GRA, we randomly selected 1600 training samples and 1000 test samples from the vehicle image dataset and implemented the VTR based on the Gabor features extracted from every patch and the proposed KSRC. Tab. 1 shows the accuracy and computing time of the VTR based on the MEGFs extracted from different Gabor kernels, where the average computing time (ACT) includes training time and test time. In Tab. 1, the used 15 Gabor features include 5 other Gabor features except the 10 MEGFs and the 20 Gabor features includes 10 other Gabor features except for the 10 MEGFs.

Tab. 2 shows the accuracy and ACT of the VTR based on other Gabor features except the MEGFs. Comparing Tab. 1 to Tab. 2, the ACT increases when the number of the used Gabor features increases; however, the accuracy, precision and recall of the VTR do not increase much. The VTR based on the MEGFs has higher accuracy, precision, and recall compared to the methods without using the MEGFs when the number of the selected Gabor features is equal. Especially when the MEGFs selected by the GRA were employed, the best performance was achieved, which has the highest accuracy, precision and recall, as well as the least computing time. Therefore, for the VTR, the Gabor feature selection based on the GRA is effective, which can not only improve the performance of the VTR but also reduce the computing time of classifiers.

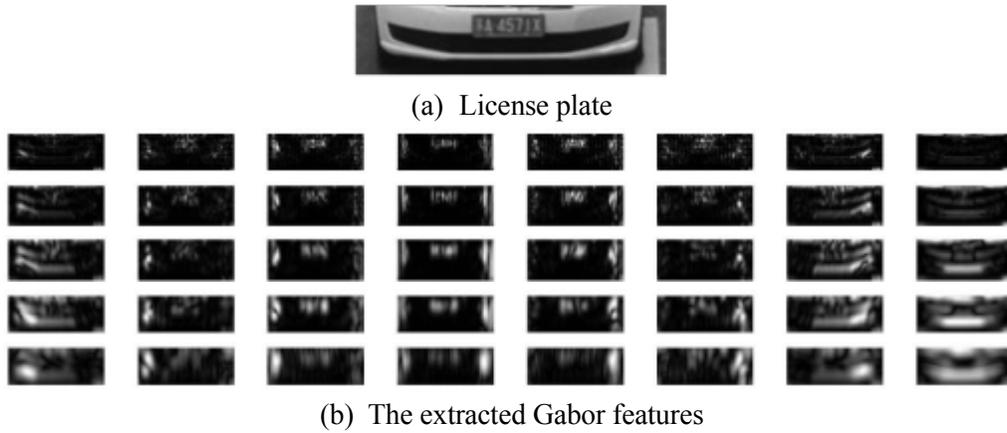


Figure 4: Gabor feature extraction

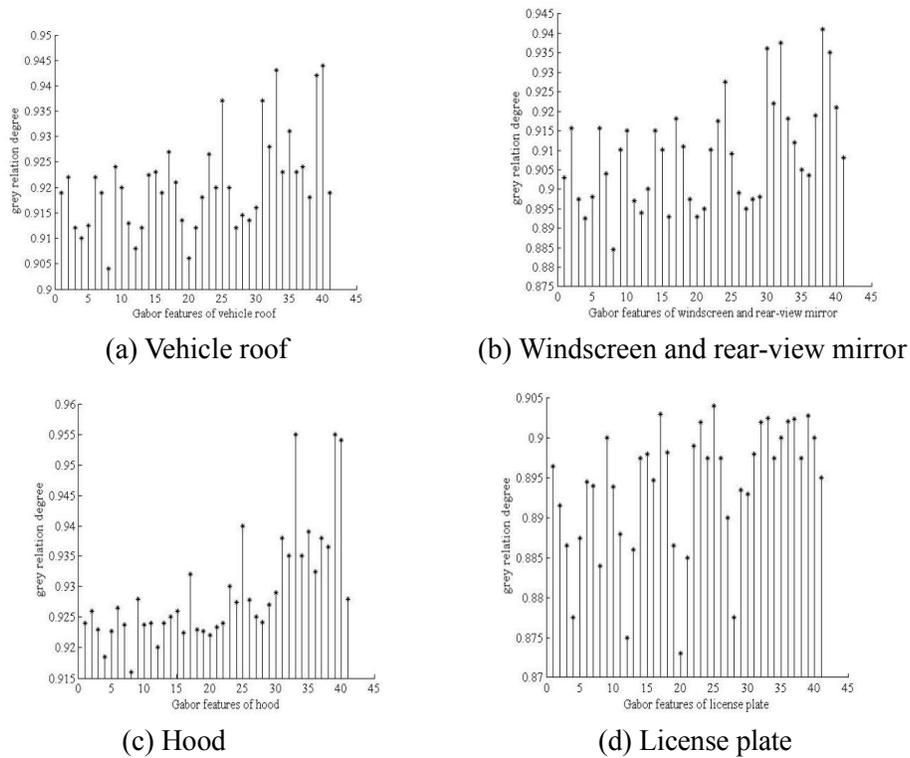


Figure 5: Grey relational degree of vehicle patches

4.2 Results of two-stage classification

4.2.1 Results of the first stage classification

For the first stage classification, we randomly selected 1600 samples as training samples,

200 samples under good illumination without occlusion as well as 200 samples under poor illumination or partial occlusion as test samples. If the vehicle type is recognized as bus or truck, the test sample is considered to be a big vehicle. Similarly, if the type is recognized as van or sedan, the test sample is considered to be a small vehicle. Tab. 3 shows the experimental results where the test samples are captured under good illumination without occlusion. Further, Tab. 4 shows the results under poor illumination or partial occlusion.

Tabs. 3 and 4 show that the VTR based on the first stage of classification has high accuracy, precision and recall, even for the test samples captured under poor illumination or partial occlusion.

Table 1: Classification results based on the MEGFs

Number of used Gabor features	Accuracy (%)	Precision (%)	Recall (%)	ACT (ms)
5	93.2	92.7	89.5	1621
10	95.8	94.4	90.9	2819
15	94.7	93.9	88.9	3820
20	93.6	91.8	87.2	5014

Table 2: Classification result without using the MEGFs

Number of used Gabor features	Accuracy (%)	Precision (%)	Recall (%)	ACT (ms)
5	87.8	86.4	85.9	1636
10	90.7	88.9	89.1	2836
15	91.0	89.8	88.6	3845
20	92.4	91.1	90.4	5114

Table 3: Result based on the first stage of classification under good illumination and non-occlusion

Vehicle type	Accuracy (%)	Precision (%)	Recall (%)
Large vehicle	98.7	97.9	97.1
Small vehicle	98.5	98.1	97.3

Table 4: Result based on the first stage of classification under poor illumination or partial occlusion

Vehicle type	Accuracy (%)	Precision (%)	Recall (%)
Large vehicle	91.7	90.8	90.6
Small vehicle	91.3	90.4	90.2

4.2.2 Results of the second stage classification

Based on the result in the first stage of classification, if the test sample is considered to be big vehicle, the sub-dataset of big vehicles will be used in the following second stage of classification. Similarly, if the test sample is recognized as a small vehicle, the sub-dataset of small vehicles will be used. We randomly selected 1600 samples as training samples including 400 bus images, 400 truck images, 400 sedan images, and 400 van images. Additionally, we also randomly selected 200 samples under good illumination without occlusion and 200 samples under poor illumination or partial occlusion as test samples from the big vehicle dataset or small vehicle dataset for further classification. Tab. 5 shows the experimental results for test samples captured under good illumination without occlusion. Tab. 6 shows the results under poor illumination or partial occlusion.

Tab. 5 and Tab. 6 shows that the second stage classification achieves high-accuracy results for the four types of vehicles. Compared to the classification results under good illumination without occlusion, the performance of the second stage classification under poor illumination or partial occlusion decreases in terms of accuracy, precision and recall; however, when partial occlusion occurs, our proposed method can still correctly recognize the vehicle type through other non-occluded key patches.

Table 5: Result based on the second stage of classification under good illumination and non-occlusion

Vehicle type	Accuracy (%)	Precision (%)	Recall (%)
Bus	96.0	95.7	95.3
Truck	96.2	95.6	94.9
Van	95.9	95.1	94.5
Sedan	95.8	94.6	95.1

Table 6: Result based on the second stage of classification under poor illumination or partial occlusion

Vehicle type	Accuracy (%)	Precision (%)	Recall (%)
Bus	90.5	88.9	89.1
Truck	90.9	89.2	89.8
Van	91.5	90.3	90.1
Sedan	92.3	91.4	90.6

4.3 Comparison with single-stage classification

To illustrate the efficacy of the two-stage scheme, we implemented single-stage classification methods using two classifiers, SKNNC and the KSRC, for comparison purposes. We randomly selected 1600 samples as training samples and 400 samples as test samples from the dataset. The single-stage classification results based on the SKNNC and edge feature, as well as the results based on the KSRC and MEGFs, are shown in Tab. 7 and Tab. 8, respectively. Comparing the results in Tabs. 7 and 8 to the results in Tab. 6, we can see that the proposed VTR method with two classification stages overpasses the single-stage classification in terms of accuracy, precision, and recall. Further research shows that the SKNNC has an excellent ability to distinguish large vehicles and small vehicles by taking position and direction into consideration. However, when the four types of vehicles are mixed, the method becomes difficult to distinguish specific vehicle type precisely. Moreover, the single-stage classification based on the KSR needs more training samples to train more classifier parameters for the involving four types of vehicles. Therefore, the performance of the single-stage classification methods will degrade in accuracy, precision, and recall compared to the proposed method based on the two-stage scheme.

Table 7: Single-stage classification results based on SKNNC and Edge feature

Vehicle type	Accuracy (%)	Precision (%)	Recall (%)
Bus	88.9	87.8	87.4
Truck	88.2	86.9	87.1
Van	87.9	86.4	85.9
Sedan	87.6	86.2	85.6

Table 8: Single-stage classification results based on KSRC and MEGFs

Vehicle type	Accuracy (%)	Precision (%)	Recall (%)
Bus	92.8	91.3	90.8
Truck	92.5	91.1	89.7
Van	92.1	90.7	89.4
Sedan	91.2	90.0	88.8

4.4 Comparison with other methods

To verify the advantages of the proposed method over other popular methods, we tested our method on the dataset in Peng et al. [Peng, Jin and Luo (2012)]. The experiments on daylight images and nightlight images were conducted respectively. Our method achieves 96.0% classification accuracy for daylight images and 91.5% for nightlight images, better than the results of other popular methods, as demonstrated in Tab. 9. In addition, we tested our proposed method on the public BIT-Vehicle dataset provided in Dong et al. [Dong, Wu, Pei et al. (2015)]. Our method achieves 90.4% classification accuracy, yet the accuracy of the method in Dong et al. [Dong, Wu, Pei et al. (2015)] reaches 88.11% only. Therefore, our

proposed method outperforms the method in Dong et al. [Dong, Wu, Pei et al. (2015)] on the BIT-Vehicle dataset.

The advantages of the proposed method are due to three reasons as follows. First, the improved Canny edge detection algorithm in the proposed method can extract the edge feature with more complete information that improves classification performance in the first stage. Second, the GRA incorporated in the proposed method helps to select more discriminative Gabor features for VTR while eliminating potential data redundancy. Third, the proposed method with a two-stage scheme leverages the advantages of the extracted edge and Gabor features in describing vehicle attributes. In details, the extracted edge feature that represents the geometrical contour of a vehicle is applied to the first stage classification only to determine whether the test sample belongs to large vehicle or small vehicle. The Gabor feature that represents the structural details of a vehicle is then applied to the second stage classification to determine whether the vehicle belongs to bus, truck, van or sedan based on the classification result from the first stage. Such a hierarchical determination process improves the accuracy and reliability of VTR.

Table 9: Comparison results between our method and other methods

Methods	Accuracy (%)	
	Daylight	Nightlight
Psyllos et al. [Psyllos, Anagnostopoulos and Kayafas (2011)]	78.3	73.3
Peng et al. [Peng, Jin and Luo (2012)]	90.0	87.6
Dong et al. [Dong and Jia (2013)]	91.3	-
Dong et al. [Dong, Wu, Pei et al. (2015)]	96.1	89.4
Ours	97.7	91.5

5 Conclusions

This paper has proposed a new two-stage VTR method combining the most effective Gabor features, which can improve classification accuracy, robustness and computation efficiency. The improved Canny edge detection algorithm is adopted to extract the edge feature of vehicle. A set of Gabor wavelet kernels with five scales and eight orientations can extract local structural details of vehicle from four partitioned key image patches. The introduced GRA can select the most effective Gabor features to reduce the dimension of input feature vector. The proposed two-stage VTR scheme can improve the reliability and robustness of VTR, where the preliminary vehicle type such as big vehicle or small vehicle is recognized based on the SKNNC and vehicle edge features. Detailed specific vehicle type, such as bus, truck, van, or sedan, is recognized based on the KSRC and the most effective Gabor features by adding a verification and correction step using

minimum residual analysis. Experimental results demonstrate the efficacy of the proposed VTR method.

Acknowledgement: We thank the three anonymous reviewers for their valuable suggestions and comments.

Funding Statement: This work is supported in part by the National Natural Science Foundation of China (Nos. 61304205 and 61502240), the Natural Science Foundation of Jiangsu Province (BK20191401), and the Innovation and Entrepreneurship Training Project of College Students (202010300290, 202010300211, 202010300116E).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- Adi, N.; Arnida, L. L.** (2015): Gabor filtering for feature extraction in real time vehicle classification system. *9th International Symposium on Image and Signal Processing and Analysis*, pp. 19-24.
- Ambardekar, A.; Nicolescu, M.; Bebis, G.; Nicolescu, M.** (2014): Vehicle classification framework: a comparative study. *Eurasip Journal on Image & Video Processing*, vol. 2014, no. 1, pp. 1-3.
- Andri, S.; Wang, C. Y.; Tai, C. Y.; Wang, J. C.** (2015): Kernel sparse representation classifier with center enhanced SPM for vehicle classification. *Computer Software & Applications Conference*, pp. 742-746.
- Chen, L. C.; Hsieh, J. W.; Yan, Y. L.; Chen, D. Y.** (2015): Vehicle make and model recognition using sparse representation and symmetrical SURFs. *Pattern Recognition*, vol. 48, no. 6, pp. 1979-1998.
- Chen, X. Y.; Gong, R. X.; Xie, L. L.; Shi, M. X.; Liu, C. L. et al.** (2017): Building regional covariance descriptors for vehicle detection. *IEEE Geoscience & Remote Sensing Letters*, vol. 14, no. 4, pp. 524-528.
- Chen, Y. T.; Wang, J.; Xia, R. L.; Zhang, Q.; Cao, Z. H. et al.** (2019): The visual object tracking algorithm research based on adaptive combination kernel. *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 12, pp. 4855-4867.
- Dong, Z.; Jia, Y. D.** (2013): Vehicle type classification using distributions of structural and appearance-based features. *IEEE International Conference on Image Processing*, pp. 4321-4324.
- Dong, Z.; Wu, Y.; Pei, M.; Jia, Y. D.** (2015): Vehicle type classification using a semisupervised convolutional neural network. *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 2247-2256.
- Fabrizia, M.; Renata, M.** (2012): An image vehicle classification method based on edge and PCA applied to blocks. *IEEE International Conference on Systems*, pp. 1688-1693.
- Fang, J.; Zhou, Y.; Yu, Y.; Du, S. D.** (2016): Fine-grained vehicle model recognition

using a coarse-to-fine convolutional neural network architecture. *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 7, pp. 1-11.

Gao, Y.; Ma, J. Y.; Yuille, A. L. (2017): Semi-supervised sparse representation based classification for face recognition with insufficient labeled samples. *IEEE Transactions on Image Processing*, vol. 26, no. 5, pp. 2545-2560.

Gu, H. Z.; Lee, S. Y. (2013): A view-invariant and anti-reflection algorithm for car body extraction and color classification. *Multimedia Tools & Applications*, vol. 65, no. 3, pp. 387-418.

He, Y.; Sang, N.; Gao, C. X.; Jun, H. (2017): Online unsupervised learning classification of pedestrian and vehicle for video surveillance. *Chinese Journal of Electronics*, vol. 26, no. 1, pp. 145-151.

Jerome, F.; Trevor, H.; Rob, T. (2010): Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, vol. 33, no. 1, pp. 1-22.

Kachach, R.; Maria, C. J. (2016): Hybrid three-dimensional and support vector machine approach for automatic vehicle tracking and classification using a single camera. *Journal of Electronic Imaging*, vol. 25, no. 3, pp. 003021-033021.

Kuang, F. J.; Zhang, S. Y.; Jin, Z.; Xu, W. H. (2015): A novel SVM by combining kernel principal component analysis and improved chaotic particle swarm optimization for intrusion detection. *Soft Computing*, vol. 19, no. 5, pp. 1187-1199.

Li, W. J.; Feng, Q. L.; Chen, J.; Hu, S. (2017): "Improved kernel results for some FPT problems based on simple observations. *Theoretical Computer Science*, vol. 657, pp. 20-27.

Li, W. J.; Liu, H. Y.; Wang, J. X.; Xiang, L. Y.; Yang, Y. J. (2019): An improved linear kernel for complementary maximal strip recovery: simpler and smaller. *Theoretical Computer Science*, vol. 786, pp. 55-66.

Li, W. J.; Zhu, B. H. (2018): A 2k-kernelization algorithm for vertex cover based on crown decomposition. *Theoretical Computer Science*, vol. 739, pp. 80-85.

Li, Y. F.; Chen, W. B.; Peeta, S.; Wang, Y. B. (2020): Platoon control of connected multi-vehicle systems under V2X communications: design and experiments. *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 5, pp. 1891-1902.

Li, Y. F.; Tang, C. C.; Peeta, S.; Wang, Y. B. (2019): Nonlinear consensus based connected vehicle platoon control incorporating car-following interactions and heterogeneous time delays. *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 6, pp. 2209-2219.

Lin, B. F.; Chan, Y. M.; Fu, L. C.; Hsiao, P.Y.; Chuang, L. A.; Huang, S. S. et al. (2012): Integrating appearance and edge features for sedan vehicle detection in the blind-spot area. *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, pp. 737-747.

Lin, M. G.; Feng, Q. L.; Chen, J. E.; Li, W. J. (2017): Partition on trees with supply and demand: Kernelization and algorithms. *Theoretical Computer Science*, vol. 657, pp. 11-19.

- Liu, W.; Wang, S. F., Gong, W. D.; Wang, Z.; Wang, R.** (2019): Machine learning based vehicle recognition and tracking using single layer lidar. *Journal of Changchun University of Science and Technology*, vol. 42, no. 3, pp. 52-64.
- Peng, Y.; Jin, J. S.; Luo, S.** (2012): Vehicle type classification using PCA with self-clustering. *Proceedings of the 2012 IEEE International Conference on Multimedia and Expo Workshops*, pp. 384-389.
- Powers, D. M. W.** (2011): Evaluation: from precision, recall and f-measure to ROC, informedness, markedness & correlation. *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37-63.
- Psyllos, A.; Anagnostopoulos, C. N.; Kayafas, E.** (2011): Vehicle model recognition from frontal view image measurements. *Computer Standards and Interfaces*, vol. 33, no. 2, pp. 142-151.
- Soon, F. C.; Khaw, H. Y.; Chuah, J. H.; Kanesan, J.** (2019): Pcanet-based convolutional neural network architecture for a vehicle model recognition system. *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 2, pp. 749-759.
- Wang, H.; Cai, Y. F.** (2016): Occluded vehicle detection with local connected deep model. *Multimedia Tools and Applications*, vol. 75, no. 15, pp. 9277-9293.
- Wang, R.; Shen, M. M.; Li, Y. P.; Gomes, S.** (2018): Multi-task joint sparse representation classification based on Fisher discrimination dictionary learning. *Computers, Materials & Continua*, vol. 57, no. 1, pp. 25-48.
- Yang, Y. J.; Shrestha, Y. R.; Li, W. J.; Guo, J.** (2018): On the kernelization of split graph problems. *Theoretical Computer Science*, vol. 734, pp. 72-82.
- Zhou, Z. Y.; Deng, T. M.; Lv, X. Y.** (2011): Study for vehicle recognition and classification based on Gabor wavelets transform & HMM. *International Conference on Consumer Electronics, Communications and Networks*, pp. 5272-5275.