Vehicle Logo Recognition by Spatial-SIFT Combined with Logistic Regression

Ruilong Chen[a], Matthew Hawes[a], Lyudmila Mihaylova[a], Jingjing Xiao[b] and Wei Liu[c]

[a]Department of Automatic Control and Systems Engineering, University of Sheffield, S1 3JD, UK
[b]School of Electronics, Electrical and Computer Engineering, University of Birmingham, B15 2TT, UK
[c]Department of Electronic and Electrical Engineering, University of Sheffield, S1 3JD, UK
{rchen3, m.hawes and l.s.mihaylova}@sheffield.ac.uk, shine636363@sina.com, w.liu@sheffield.ac.uk

***Abstract*—An efficient recognition framework requires both good feature representation and effective classification methods. This paper proposes such a framework based on a spatial Scale Invariant Feature Transform (SIFT) combined with a logistic regression classifier. The performance of the proposed framework is compared to that of state-of-the-art methods based on the Histogram of Orientation Gradients, SIFT features, Support Vector Machine and K-Nearest Neighbours classifiers. By testing with the largest vehicle logo data-set, it is shown that the proposed framework can achieve a classification accuracy of 99.93%, the best among all studied methods. Moreover, the proposed framework shows robustness when noise is added in both training and testing images.**

I. Introduction

Recognizing vehicle logos is important in Intelligent Transportation Systems as the vehicle logo is one of the most distinguishable marks on a vehicle [1], and can assist in vehicle identification [2]. For instance, vehicle logo recognition can detect fraudulent plates if the combination does not match the data stored on the police security database [3]. As a result, this gives a more robust vehicle identification system. In addition, vehicle logo recognition is also very useful for commercial investigations [4] and document retrieval [5].

Hand crafted features are often used to represent the content in an image. There are global features which take all pixels into account, such as the Histogram of Oriented Gradients (HOG) feature [6], and local features which are only interested in a few significant points, such as the SIFT feature [7]. Both global and local features are explored in vehicle logo recognition applications [1], [8], [9], [10], [2], [11]. In general, local features are more often used, as global features are sensitive to illumination and scale changes, background noises and rotations, whereas local features tend to be more robust under these severe conditions [12].

As well as good feature representation schemes, for high recognition accuracy we also require good classification methods. In literature, the K-Nearest Neighbours (KNN) [13] classifier is often used as a base line [14], [15], whereas the more advanced Support Vector Machine (SVM) classifier is often a more appropriate choice [1], [8], [9], [14], [4], [16].

In this paper, we propose a classification framework based on logistic regression (LR) and a spatial SIFT feature representation scheme. The LR explores the confidence level of the classification decision and the spatial SIFT feature representation adds the geographic information of SIFT features. This framework is compared with methods based on HOG, SIFT, SVM, and KNN to verify its effectiveness for both clean and noisy images.

The rest of this paper is organized as follows. In Section II, we explain how the HOG feature (global feature) and the SIFT feature (local feature) are combined with the Bag of Words (BOW) representation model. In Section III. A, we explain how we implement the pyramid idea based on the SIFT feature. Section III. B introduces how the logistic regression is employed in order to solve the multi-classification problem. Experimental result and discussions are presented in Section IV and the Section V summarises the work.

II. Related works

A good recognition system needs good features to represent the image. In the following two state-of-the-art feature methods are introduced, namely the HOG feature and the SIFT feature. Using the HOG feature, all images are represented by a vector of the same length and therefore, they can be classified directly. For local features such as the SIFT feature, the number of features is normally different. Therefore, the BOW representation model is required prior to classification.

A. HOG features

HOG calculates the horizontal gradient G_x and the vertical gradient G_y on every pixel in the image using a 1-D filter [-1, 0, 1] [6], [17]

$$G_{x(i,j)} = f(i+1,j) - f(i-1,j), \tag{1}$$

$$G_{y(i,j)} = f(i,j+1) - f(i,j-1), \tag{2}$$

where $f(i,j)$ is the intensity value at pixel location (i,j). Then the horizontal gradient and vertical gradient can be used to calculate the orientation of gradient $\theta(i,j)$ and the magnitude of gradient $H(i,j)$ for every pixel in the image

$$\theta(i,j) = arctan(G_{x(i,j)}/G_{y(i,j)}), \tag{3}$$

$$H(i,j) = \sqrt{G^2_{x(i,j)} + G^2_{y(i,j)}}. \tag{4}$$

The image is then divided into cells and blocks, where a cell is made up from a few pixels and a block is made up from a few cells. Each block can be represented as a histogram using the quantized orientations as the histogram bins and the magnitude as the weights. For each histogram the orientations are quantised into bins evenly spaced over the full angular range. The HOG feature is the concatenation of the histogram vectors of all blocks.

B. SIFT features

Local features are often more effective and more robust than global features [12]. Compared with global features which use information from the whole image, local features are only interested in distinctive information from set point regions in the image. As a result, local feature methods need to detect which pixels are of interest and then describe these pixels using their neighbourhood areas.

Among local features, the SIFT feature is the most successful one. The SIFT feature [7] is invariant to scale, rotation, affine distortion, and noise. It detects a set of interest points and then calculates the histogram of gradients in a window centered around them. In the interest points detection process, different Gaussian filters $G(x,y,k\sigma)$ are convolved with the original image to get smoothed images $L(x,y,k\sigma)$. Then the Difference of Gaussians (DOG) $D(x,y,\sigma)$ is generated by calculating the differences between these Gaussian smoothed images, which is defined as:

$$D(x,y,\sigma) = L(x,y,k\sigma) - L(x,y,\sigma), \tag{5}$$

where k is a constant multiplicative factor usually set to $\sqrt{2}$ [7]. $L(x,y,\sigma)$ and $L(x,y,k\sigma)$ are produced from variant scale Gaussian filters convolved with the input image, $I(x,y)$:

$$L(x,y,\sigma) = G(x,y,\sigma) * I(x,y), \tag{6}$$

where the Gaussian filter is defined as:

$$G(x,y,\sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}. \tag{7}$$

The DOG is not only applied to the original image but also the up-sampled and down-sampled images in order to be scale invariant. The potential interest points are extrema among their neighbours in the DOG maps. All the extrema are then revalued in order to make interest points more robust by rejecting the less significant extrema. Finally the locations of remaining extrema are used as the locations of interest points.

After the location of an interest point has been detected, its neighbourhood area is chosen in the corresponding scale. In order to make each interest point invariant to rotation changes, all gradient orientations are rotated relative to the main orientation of the local area. Any other orientation, which is within 80% magnitude of the main orientation, will be used to create another descriptor with respect to that orientation. It makes each local interest point can have multiple descriptors. After the orientation assignment process, an area centered on the interest point is chosen. A similar process of the HOG model is then applied within the area. Unlike in the HOG process, in SIFT all weights for orientations are decided by both the magnitude of gradients and a Gaussian kernel which is centered on the interest point. All histograms are then concatenated into a vector of fixed length. Finally, the vector is normalized in order to be invariant to illumination changes, and the normalized vector is the SIFT descriptor.

For different images there may be a different number of SIFT descriptors, as the number of interest points is determined by the extrema and the number of extrema does not have to be the same in every image. In other words, images are represented by matrices with different sizes. Therefore, they cannot be directly classified. Besides, directly comparing each descriptor from testing images with all the descriptors in the training dataset seems impractical when the training dataset is huge [18]. In order to solve this problem, the BOW representation model is required prior to classification.

C. Bag of Words

Csurka et al [18] proposed the BOW model to represent an image by a feature histogram, which is efficient in terms of computational cost and practical implementation. It is also often used in vehicle logo recognition [19], [20]. The BOW model consists of two main steps: the dictionary generation process by k-means clustering [21] and the histogram representation process.

k-means clustering is an unsupervised vector quantization algorithm. It clusters n observations into k clustering centroids by allocating all the observations into its nearest centroid. The algorithm involves four steps:

1) Randomly choose k points as the initial group centroids in the training data.
2) Assign all the training data points to its nearest centroid.
3) When all data points have been assigned, find the center of each group and assign it as the new centroid.

4) Repeat steps 2 and 3 until none of the centroids changes any more.

By the k-means clustering method, all the training features are used to generate a dictionary which is made up of k 'words' (centroids) and each 'word' has the same dimension as a SIFT feature vector. For an image which consists of a few local descriptors, each descriptor can find its closest 'word' from the dictionary, where the closest distance is defined as the minimal l_2 distance [22]. If a descriptor has found its nearest 'word' in the dictionary, the number of occurrences of this 'word' will have increased by 1. The BOW model can represent an image as a histogram by using each 'word' in the dictionary as a histogram bin and the occurring frequency of each 'word' as its magnitude [18]. The normalized vector is the final histogram vector.

III. Proposed Framework for Vehicle Logo Recognition

The proposed vehicle logo recognition framework which combines the spatial SIFT feature with logistic regression classification is shown in figure 1. This section introduces spatial SIFT and logistic regression, as the remaining stages having been detailed in the previous section in this paper.

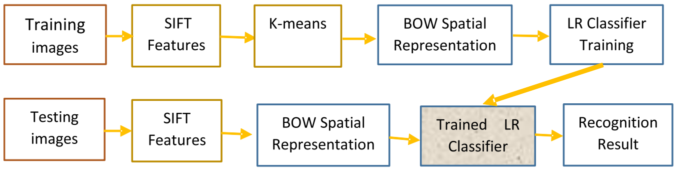

Fig. 1: Recognition framework by using local features.

A. Spatial SIFT

In the BOW model, the magnitude of each 'word' in the histogram is only decided by its occurring frequency in the image. Where the feature was originally from in the image does not influence the histogram. Therefore it does not take the geographic information into consideration. The geographic information of the interest points is often deliberately avoided in order to ensure that interest points at different locations can be matched, making the process invariant to changes in interest points locations. However, vehicle logos often occupy the entire of the training and testing images after a segmentation process. The geometric information might be useful in such a case. For example, the 'V' is always above the 'W' in a detected Volkswagen logo image and using such information potentially can give a more accurate classification.

Lazebnik et al [23] proposed the idea of partitioning the image into sub-regions and then using the BOW model over each sub-region for natural images. Specifically, the original image is firstly partitioned into 4 sub-regions, then into 16 sub-regions in the next level and so on. The BOW model is applied over each sub-region and the final feature is formed by concatenating the histograms from the original image and all the sub-regions. The result is a pyramid-like structure, where each level as you move down the pyramid is focused on a smaller region of the image. Each level is often called a pyramid scale.

This pyramid idea has been applied in vehicle logo recognition tasks by using the Dense-SIFT global descriptor [4], [19]. However, the Dense-SIFT feature takes all pixels in the original map as interest points, which makes the interest points not robust as a lack of feature detection process [24]. Instead, here we propose using the pyramid idea with the SIFT local feature descriptor for vehicle logo recognition.

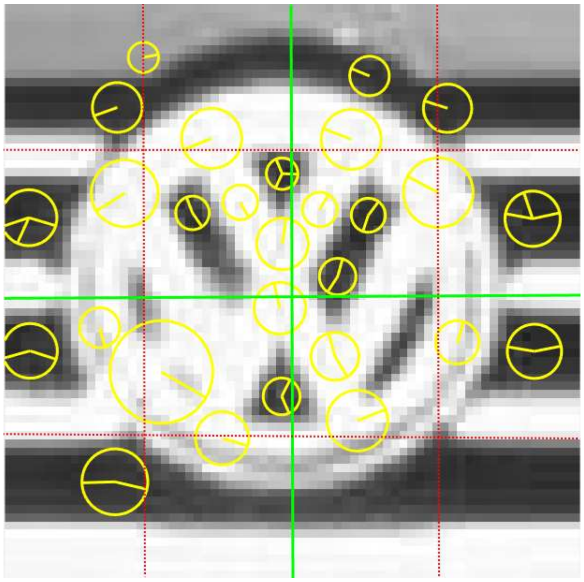

Fig. 2: An example of spatial pyramid interest points. The center of the yellow circles are locations of SIFT features in their corresponding maps and the yellow bars represent the main orientations. Since the interest points are from different DOG maps, the size of the yellow circles varies.

Figure 2 illustrates the pyramid partition of an image. By using the BOW representation model, the original image can be represented by a histogram of k dimensions (k is defined by k-means algorithm) in the first pyramid scale; then, the image is divided into 4 sub-regions and 16 sub-regions in the second and third pyramid scales, respectively. The BOW model is applied over each region to obtain histogram vectors and all these histogram vectors generated from both original scale and sub-scales are concatenated into a histogram vector to represent the image.

Figure 3 shows an example of how the BOW model represent the image in figure 2, using the SIFT feature and the spatial-SIFT feature. For illustration purpose, k=50 is used in the k-means clustering and 2 pyramid levels are used for the spatial-SIFT feature. By using the SIFT feature, the BOW is only applied to the original image therefore the image is represented by a histogram vector of length 50 (figure 3 (a)). However, using spatial SIFT, BOW is applied to both the original image and the sub-regions. Hence, the image is represented by a vector of length 250 (figure 3 (b)). Therefore, as both SIFT and spatial SIFT are sharing the same dictionary, the SIFT vector forms the first portion of the spatial SIFT vector.

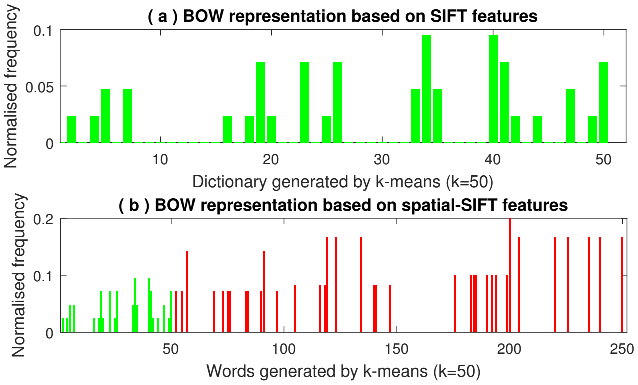

Fig. 3: The BOW representation for the image in figure 2 based on the SIFT feature (a) and the spatial SIFT feature (b).

B. Logistic regression

Unlike SVM and KNN, the LR has not previously been applied to vehicle logos recognition. Compared with the conventional SVM and KNN which only classify data into corresponding classes, LR explores the confidence level of the decision that the data has been correctly classified [25]. The following gives an introduction about how logistic regression can be used in multi-class classification in order to solve the multi-class vehicle logo recognition problem.

Given a training data $(\mathbf{x}, y)$, where $\mathbf{x} \in \mathbb{R}^{M\times 1}$, in linear regression, we use the linear function:

$$y = \mathbf{w}^T\mathbf{x} + b, \tag{8}$$

where $\mathbf{w} \in \mathbb{R}^{M\times 1}$ is the weight vector and the scalar b is the bias associated with the linear regression. Starting with the binary classification where y is a scalar which can either be ‘1’ (positive) or ‘0’ (negative). Using the ‘logistic’ function $f(x) = 1/(1 + e^{-x})$, the probability that the training point belongs to class ‘1’ can be expressed by:

$$\pi = p(y = 1|\mathbf{x}, \mathbf{w}, b) = f(\mathbf{w}^T\mathbf{x} + b) = \frac{1}{1 + e^{-(\mathbf{w}^T\mathbf{x}+b)}}. \tag{9}$$

Therefore, the probability of a negative outcome is $1 - \pi$:

$$p(y = 0|\mathbf{x}, \mathbf{w}, b) = 1 - \pi = \frac{e^{-(\mathbf{w}^T\mathbf{x}+b)}}{1 + e^{-(\mathbf{w}^T\mathbf{x}+b)}}. \tag{10}$$

Assuming that we have N independent training data $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_N, y_N)$, a Bernoulli distribution can be used to form the likelihood function for the i^{th} point by combining Equations (9) and (10), which gives:

$$p(y_i|\mathbf{x}_i, \mathbf{w}, b) = \pi_i^{y_i}(1 - \pi_i)^{1-y_i}, \tag{11}$$

where π_i represents the probability that the i^{th} point belongs to the positive class. The likelihood of all the training data is therefore given by the product:

$$p(\mathbf{y}|\mathbf{w}, \mathbf{X}, b) = \prod_{i=1}^{N} \pi_i^{y_i}(1 - \pi_i)^{1-y_i}, \tag{12}$$

where $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N)^T$ is the training dataset and $\mathbf{y} \in \mathbb{R}^{N\times 1}$ is a vector representing all the training labels. Maximising the likelihood in Eq. (12) is equivalent to minimising the negative of its log likelihood, i.e.

$$\begin{aligned} &E = -log\ p(\mathbf{y}|\mathbf{w}, \mathbf{X}, b) \\ &= -\sum_{i=1}^{N} y_i log\pi_i - \sum_{i=1}^{N}(1 - y_i)log(1 - \pi_i) \\ &= -\sum_{i=1}^{N} y_i log\ f(\mathbf{w}^T\mathbf{x}_i + b) - \sum_{i=1}^{N}(1 - y_i)log(1 - f(\mathbf{w}^T\mathbf{x}_i + b)). \end{aligned} \tag{13}$$

In order to minimize Eq. (13), we take the gradient with respect to $\mathbf{w}$ and b respectively and substitute $f'(x) = f(x)(1 - f(x))$:

$$\begin{aligned} &\frac{dE}{d\mathbf{w}} = -\sum_{i=1}^{N} \frac{y_i}{f(\mathbf{w}^T\mathbf{x}_i + b)} f'\boldsymbol{x}_i + \sum_{i=1}^{N} \frac{1 - y_i}{1 - f(\mathbf{w}^T\mathbf{x}_i + b)} f'\boldsymbol{x}_i \\ &= -\sum_{i=1}^{N} y_i(1 - f(\mathbf{w}^T\mathbf{x}_i + b))\mathbf{x}_i + \sum_{i=1}^{N}(1 - y_i)f(\mathbf{w}^T\mathbf{x}_i + b)\mathbf{x}_i \\ &= -\sum_{i=1}^{N}(y_i - f(\mathbf{w}^T\mathbf{x}_i + b))\mathbf{x}_i, \end{aligned} \tag{14}$$

here f' represents the partial derivative of $f(\mathbf{w}^T\mathbf{x}_i + b)$ with respect to $\mathbf{w}$. In the same way take the gradient with respect to b:

$$\frac{dE}{d\ b} = -\sum_{i=1}^{N}(y_i - f(\mathbf{w}^T\mathbf{x}_i + b)). \tag{15}$$

Equations (14) and (15) are optimization problems which are usually solved by gradient descent method such as stochastic gradient descent [26] and Newton’s method [27]. For a new testing point $\mathbf{x}^*$, the probability that it belongs to the *positive* class is:

$$p(y^* = 1|\mathbf{x}^*, \mathbf{w}, b) = \frac{1}{1 + e^{-(\mathbf{w}^T\mathbf{x}^*+b)}}, \tag{16}$$

and the probability that it belongs to the *negative* class is therefore:

$$p(y^* = 0|\mathbf{x}^*, \mathbf{w}, b) = 1 - p(y^* = 1|\mathbf{x}^*, \mathbf{w}, b), \tag{17}$$

where y^* represents the predicted label for the testing point. Hence, the testing data can be allocated into the class which has a higher probability. In practice an l_2 regularizer is added in the object function (shown in Equation (13)) in order to avoid over-fitting [28]. The regularised object function is:

$$\hat{E} = -log\ p(\mathbf{y}|\mathbf{w}, \mathbf{X}, b) + \frac{\lambda}{2}(||\mathbf{w}||_2^2 + b^2), \tag{18}$$

where $||\cdot||_2$ denotes l_2-norm and λ is the weight controlling the importance of the regularization term.

The logistic regression in binary classification can be easily extended to multi-classification. Given the training dataset $\mathbf{X}$ from K categories $y_i \in 1, 2, \cdots, K$. In multi-classification, the probability of $p(y_i = k|\mathbf{x}_i)$ for each $k = (1, 2, \cdots, K)$ can be denoted as:

$$\begin{bmatrix} p(y_i{=}1|\mathbf{x}_i,\mathbf{W},\mathbf{b}) \\ p(y_i{=}2|\mathbf{x}_i,\mathbf{W},\mathbf{b}) \\ \vdots \\ p(y_i{=}K|\mathbf{x}_i,\mathbf{W},\mathbf{b}) \end{bmatrix} = \frac{1}{\sum_{j=1}^{K} e^{(\mathbf{w}_j^T\mathbf{x}_i + b_j)}} \begin{bmatrix} e^{(\mathbf{w}_1^T\mathbf{x}_i + b_1)} \\ e^{(\mathbf{w}_2^T\mathbf{x}_i + b_2)} \\ \vdots \\ e^{(\mathbf{w}_K^T\mathbf{x}_i + b_K)} \end{bmatrix}, \tag{19}$$

where $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_K)$ is a matrix consisting of the weights and $\mathbf{b} = (b_1, b_2, \cdots, b_K)$ is the bias of the multi-class logistic regression models. The term $\frac{1}{\sum_{j=1}^{K} e^{(\mathbf{w}_j^T\mathbf{x}_i + b_j)}}$ normalizes the distribution so that all the probabilities sum up to one.

Here we could use an indicator function:

$$g(a) = \begin{cases} 1 & if \quad a = True\ statement, \\ 0 & Otherwise. \end{cases} \tag{20}$$

Therefore, the object function in Eq. (13) is adapted to:

$$\begin{aligned} \hat{E} = &\sum_{i=1}^{N}\sum_{k=1}^{K} g(y_i = k) \\ &\times \left\{ -log\left(\frac{e^{(\mathbf{w}_k^T\mathbf{x}_i + b_k)}}{\sum_{j=1}^{K} e^{(\mathbf{w}_j^T\mathbf{x}_i + b_j)}} \right) + \frac{\lambda}{2}(||\mathbf{w}_k||_2^2 + b_k^2) \right\}, \end{aligned} \tag{21}$$

which is minimised to estimate $\mathbf{w}_k$ and b_k in the same way as in binary classification. For a testing data point $\mathbf{x}^*$, the probability that its label y^* equals k is :

$$p(y^* = k|\mathbf{x}^*, \mathbf{W}, \mathbf{b}) = \frac{e^{(\mathbf{w}_k^T\mathbf{x}^* + b_k)}}{\sum_{j=1}^{K} e^{(\mathbf{w}_j^T\mathbf{x}^* + b_j)}}, \tag{22}$$

and this incoming testing data is assigned to the class which has the highest probability.

IV. Performance Evaluation

In this section we use the open dataset provided by Huang et al [29] to evaluate the performance of our framework. This dataset is currently the biggest available vehicle logo dataset; it has 10 categories and each category contains 1000 training images and 150 testing images. All images have a size of 70×70 pixels. Figure 4 shows an example of these 10 vehicle categories by randomly choosing one image from each category in the training dataset and figure 5 shows some challenging test images which can be easily mis-classified.

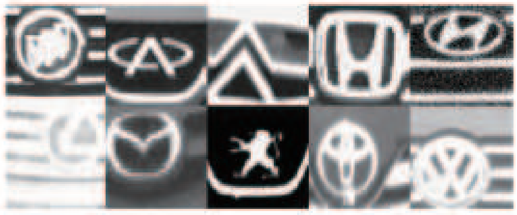

Fig. 4: Vehicle logo dataset

Fig. 5: Examples of some challenge images in the testing dataset.

The performance evaluation is conducted in Matlab on a computer with the following specification: I5, 3.4G Quad-core, and 8G memory. The open source library VLFeat [30] is used for SIFT feature extraction and LIBSVM toolbox [31] is used for SVM classification. The following result shows the performance of the HOG, SIFT and spatial SIFT features when they are combined with different classifiers such as the SVM, LR and KNN. Different levels of noise are added in order to examine the robustness of the proposed framework.

A. The HOG feature

The framework for HOG features is only made up of feature and classification, as no k-means process is needed. Three classifiers are used in this section and the following for feature classification, which are the KNN, SVM and LR. K is setted to 5 to be more robust against noisy data [32]; the SVM uses the default Radial Basis Function (RBF) kernel in LIBSVM and $\lambda = 0.1$ is setted in the LR classifier.

TABLE I: Performance of HOG by using different classifiers.

HOG features

Classifier	SVM	LR	KNN
Acc (%)	88.40	**97.53**	95.67
Misclassified images from 1500 testing images	174	**37.05**	64.95

From Table I we can see that LR outperforms SVM and KNN in terms of classification accuracy. This validates the use of LR in vehicle logo recognition.

TABLE II: Classification accuracies ($\mu \pm \sigma$) on 1500 testing images using SIFT features, according to different dictionary sizes in the k-means process (30 runs).

SIFT features

K	50	100	200	300	400	500	600	700	800	900
SVM (%)	88.09 ± 1.01	92.87 ± 0.55	95.89 ± 0.61	97.09 ± 0.46	97.64 ± 0.27	97.77 ± 0.25	97.96 ± 0.26	98 ± 0.19	98 ± 0.25	98.05 ± 0.19
Misclassified images	178.65	106.95	61.65	43.65	35.40	33.45	30.60	30	30	29.25
LR (%)	88.76 ± 1.13	95.18 ± 0.62	98.56 ± 0.32	99.11 ± 0.22	99.37 ± 0.16	99.54 ± 0.12	99.63 ± 0.11	99.70 ± 0.10	99.70 ± 0.13	**99.76 ± 0.14**
Misclassified images	170.40	72.30	21.60	13.35	9.45	6.90	5.55	4.50	4.50	**3.60**
KNN (%)	96.55 ± 2.33	97.93 ± 0.37	98.55 ± 0.33	98.60 ± 0.26	98.46 ± 0.28	98.66 ± 0.25	98.68 ± 0.27	98.73 ± 0.29	98.69 ± 0.25	98.81 ± 0.24
Misclassified images	51.75	31.05	21.75	21	23.10	20.10	20.10	19.05	19.65	17.85

TABLE III: Classification accuracies ($\mu \pm \sigma$) on 1500 testing images by using spatial-SIFT features, according to different dictionary sizes in the the k-means process (30 runs).

Spatial-SIFT features

K	50	100	200	300	400	500	600	700	800	900
SVM (%)	89.52 ± 0.88	94.26 ± 0.59	96.36 ± 0.37	96.84 ± 0.32	97.04 ± 0.38	97.10 ± 0.26	97.12 ± 0.25	97.08 ± 0.31	97.08 ± 0.26	97.20 ± 0.24
Misclassified images	157.20	86.10	54.60	47.40	44.40	43.50	43.20	43.80	43.80	42
LR (%)	95.34 ± 0.63	98.55 ± 0.25	99.55 ± 0.16	99.71 ± 0.12	99.81 ± 0.13	99.87 ± 0.09	99.86 ± 0.08	99.90 ± 0.09	99.92 ± 0.08	**99.93 ± 0.07**
Misclassified images	69.90	21.75	6.75	4.35	2.85	1.95	2.10	1.50	1.20	**1.05**
KNN (%)	92.48 ± 0.45	93.11 ± 0.60	93.56 ± 0.69	93.85 ± 0.59	94.12 ± 0.53	97.14 ± 0.49	96.82 ± 0.68	97.02 ± 0.60	96.98 ± 0.35	96.77 ± 0.38
Misclassified images	112.80	103.35	96.60	92.25	88.20	42.90	47.70	44.70	45.30	48.45

B. The SIFT feature

For local features such as SIFT and spatial-SIFT, we expect the dictionary size will influence the performance in terms of accuracy. Hence, the SVM, KNN and LR with 10 different dictionary sizes are tested. Centroids of k-means clustering are randomly initialised which results different outcomes with each run, therefore the experiments are conducted 30 times and the mean results with variances are presented.

From Table II, we can see that increasing dictionary size improves the classification accuracy. This is because the feature naturally contains more information in a higher dimensional space. Among these three classifiers, the LR classifier always outperforms the SVM, while the KNN sometimes works better than the rest when the dictionary size is smaller than 300. However, when the dimension increases, the improvement of KNN is not as obvious as for both the SVM and LR. Furthermore, the variance of accuracy achieved when using the LR is smaller than for both SVM and KNN. This indicates that the performance of the LR classifier is more stable. By combining SIFT features with the LR classifier, we have obtained the recognition accuracy 99.76±0.14, which is higher than the previous highest accuracy achieved by PCA-CNN (99.13 ± 0.24) [29].

C. The spatial-SIFT feature

The pyramid level is set to 2 as the number of SIFT features is very limited in such low resolution images. The recognition accuracies using the spatial-SIFT feature and different classifiers are shown in Table III. For both SVM and LR, the result indicates that the spatial-SIFT feature outperforms the SIFT feature no matter how large the dictionary is. However, for KNN, using spatial-SIFT has the opposite effect and accuracy is reduced. This is because the feature has been extended into a high dimension space by segmenting the image into pyramid sub-regions and studies show that KNN is not sufficient when it is applied in a high dimensional space [33]. Since the pyramid SIFT outperforms the SIFT by using both the LR and the SVM, we can conclude that the spatial-SIFT feature outperforms the SIFT in terms of accuracy, because the reduced accuracy by the fact that using KNN can be explained by the data has been extended to a high dimension. Furthermore, the pyramid SIFT combined with the LR classifier has achieved a further improvement in accuracy over that for LR combined with SIFT features.

D. Computational costs

TABLE IV: Computational costs by using different features with the LR classifier on 10000 training images and 1500 testing images.

Features	**HOG**	**SIFT**	**spatial SIFT**
Acc (%)	97.53	99.11	99.71
Misclassified images	37.05	13.35	4.35
Time-whole-process (s)	190	571	967
Time-per-test (s)	0.06	0.08	0.23

We have compared the efficiency of these features using the LR classifier. In this subsection and the one that follows we only use LR as the classifier. This is because the results up to this point indicate that it is the most accurate of the classifiers. We set k=300 as a compromise between computational cost and accuracy. It is a dilemma here as increasing the dictionary size also increases the computational cost. For example, by the proposed framework, recognising all the testing images needs 847 and 1291 seconds when $k = 200$ and $k = 400$ respectively with its accuracy increased by 0.27%. The result in Table IV indicates that spatial SIFT obtained the highest

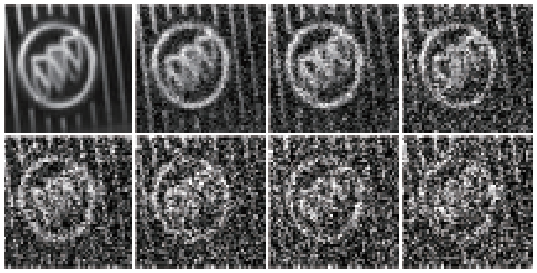

Fig. 6: An example of a training image and the effect by adding Gaussian noise with zero means and variance values 0.02, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3 from left to right respectively.

accuracy (99.71%) compared with the SIFT (99.11%) and the HOG (97.53%), respectively.

E. Robustness to noise

In practice, we would not expect to always have clear logos in the images. As a result, here noises are added to the images in order to test the robustness of these features with the LR classifier. The noise is Gaussian with zero mean and differing levels of variance. Since the Gaussian noise is random, we run all experiments for 10 times and choose their mean values. Figure 6 shows an original training example and the effects by adding noise with increasing variances. Normally an image is highly contaminated if the Gaussian noise variance is above 0.2. The noise is added to the training and testing images separately with variances given by σ_{train} and σ_{test}, respectively.

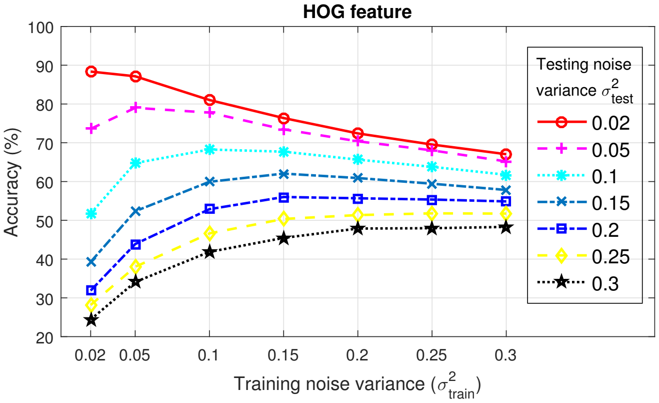

Fig. 7: Accuracy of recognition by using the HOG feature.

Figure 7 illustrates how different noise variance levels for both training and testing images influence the accuracy when using the HOG feature. Without surprise, adding noise in the image decreases the accuracy compared to the noise free case. Generally speaking, when the noise variance in the training set is fixed, the higher the noise variance added to the test images is, the lower the accuracy will be. For example, σ^2_{test}=0.02 always outperforms σ^2_{test}=0.3 in terms of accuracy. However, when the noise in the testing images is fixed, the highest accuracy tends to be found when the training images have similar noise variance levels. For instance, the highest accuracy for σ^2_{test}=0.05 is found when σ^2_{train}=0.05; on the contrary, the model trained by clearer training images (when σ^2_{train}=0.02) gives a less accurate recognition result. As a result, a higher accuracy can be achieved by matching σ^2_{train} to σ^2_{test}.

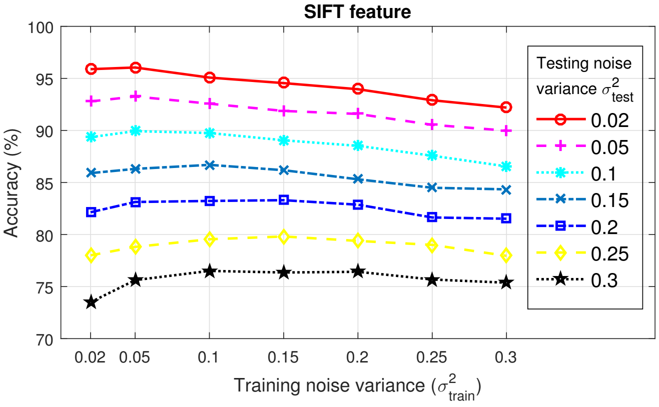

Fig. 8: Accuracy of recognition by using the SIFT feature.

Compared with the HOG feature, the SIFT feature is more robust to noise as shown in figure 8. The main reason why some misclassifications occur in extreme noise scenarios is that no SIFT features are detected. However, this doesn't always happen, meaning a good overall recognition accuracy is achieved.

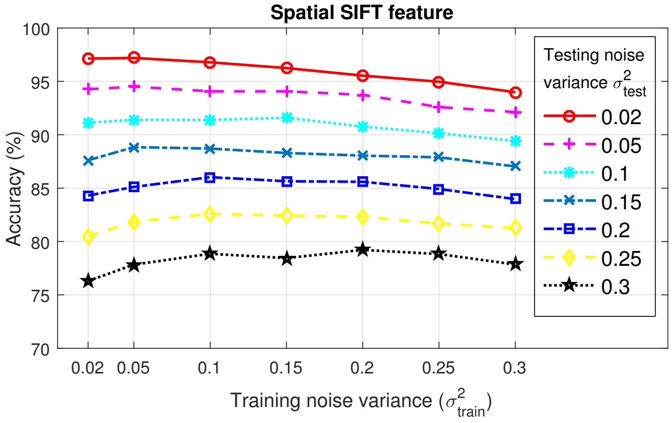

Fig. 9: Accuracy of recognition using the spatial-SIFT feature.

As shown in figure 9, the spatial-SIFT feature gives the highest accuracy when compared with the SIFT feature and the HOG feature. All accuracy results in figure 9 are above the corresponding ones in figure 8. This improvement shows that the geographic information included in spatial-SIFT has resulted in a more robust performance than for SIFT.

V. Summary

In this work, a framework based on spatial-SIFT features combined with logistic regression has been proposed for vehicle logo recognition. The spatial-SIFT features which include the geographic knowledge of SIFT features are more robust than both SIFT and HOG in both noise-free and noisy cases. Three classifiers (SVM, LR, and KNN) were tested and the LR shows an overall higher accuracy than both the SVM and KNN. The proposed framework achieved an recognition accuracy of 99.93%, which exceeded the previous record.

References

[1] D. Llorca, R. Arroyo, and M. Sotelo, "Vehicle logo recognition in traffic images using hog features and svm," in *Proc. Intelligent Transportation Systems*. IEEE, 2013, pp. 2229–2234.

[2] A. P. Psyllos, C.-N. E. Anagnostopoulos, and E. Kayafas, "Vehicle logo recognition using a sift-based enhanced matching scheme," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 2, pp. 322–328, 2010.

[3] L. Figueiredo, I. Jesus, J. T. Machado, J. Ferreira, and J. M. De Carvalho, "Towards the development of intelligent transportation systems," in *Proc. Intelligent transportation systems*, vol. 88, 2001, pp. 1206–1211.

[4] Y. Ou, H. Zheng, S. Chen, and J. Chen, "Vehicle logo recognition based on a weighted spatial pyramid framework," in *Proc. IEEE 17th International Conf. on Intelligent Transportation Systems*, 2014, pp. 1238–1244.

[5] Z. Zhang, X. Wang, W. Anwar, and Z. L. Jiang, "A comparison of moments-based logo recognition methods," in *Proc. Abstract and Applied Analysis*, vol. 2014. Hindawi Publishing Corporation, 2014.

[6] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2005.

[7] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[8] H. Pan and B. Zhang, "An integrative approach to accurate vehicle logo detection," *Journal of Electrical and Computer Engineering*, vol. 2013, p. 18, 2013.

[9] Q. Sun, X. Lu, L. Chen, and H. Hu, "An improved vehicle logo recognition method for road surveillance images," in *Proc. Seventh International Symposium on Computational Intelligence and Design*, vol. 1, Dec 2014, pp. 373–376.

[10] H. Yang, L. Zhai, L. Li, Z. Liu, Y. Luo, Y. Wang, H. Lai, and M. Guan, "An efficient vehicle model recognition method," *Journal of Software*, vol. 8, no. 8, pp. 1952–1959, 2013.

[11] R. Lipikorn, N. Cooharojananone, S. Kijsupapaisan, and T. Inchayanunth, "Vehicle logo recognition based on interior structure using sift descriptor and neural network," in *Proc. International Conf. on Information Science, Electronics and Electrical Engineering*, vol. 3, April 2014, pp. 1595–1599.

[12] C. Wallraven, B. Caputo, and A. Graf, "Recognition with local features: the kernel recipe," in *Proc. Ninth IEEE International Conf. on Computer Vision*, Oct 2003, pp. 257–264 vol.1.

[13] T. M. Cover and P. E. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[14] J. Xiao, W. Xiang, and Y. Liu, "Vehicle logo recognition by weighted multi-class support vector machine ensembles based on sharpness histogram features," *IET, Image Processing*, vol. 9, no. 7, pp. 527–534, 2015.

[15] Q. Sun, X. Lu, L. Chen, and H. Hu, "An improved vehicle logo recognition method for road surveillance images," in *Proc. IEEE Seventh International Symposium on Computational Intelligence and Design*, vol. 1, 2014, pp. 373–376.

[16] B. Zadrozny and C. Elkan, "Transforming classifier scores into accurate multiclass probability estimates," in *Proc. of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2002, pp. 694–699.

[17] O. Ludwig, D. Delgado, V. Goncalves, and U. Nunes, "Trainable classifier-fusion schemes: An application to pedestrian detection," in *Proc. 12th International IEEE Conf. on Intelligent Transportation Systems*, Oct 2009, pp. 1–6.

[18] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Proc. Workshop on statistical learning in computer vision, ECCV*, vol. 1, no. 1-22. Prague, 2004, pp. 1–2.

[19] S. Yu, S. Zheng, H. Yang, and L. Liang, "Vehicle logo recognition based on bag-of-words," in *Proc. 10th IEEE International Conf. on Advanced Video and Signal Based Surveillance*. IEEE, 2013, pp. 353–358.

[20] Y. Kalantidis, L. G. Pueyo, M. Trevisiol, R. van Zwol, and Y. Avrithis, "Scalable triangulation-based logo recognition," in *Proc. of the 1st ACM International Conf. on Multimedia Retrieval*. ACM, 2011, p. 20.

[21] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*. Berkeley, Calif.: University of California Press, 1967, pp. 281–297.

[22] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *Proc. Ninth IEEE International Conf. on Computer Vision*. IEEE, 2003, pp. 1470–1477.

[23] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, vol. 2. IEEE, 2006, pp. 2169–2178.

[24] T. Hassner, V. Mayzels, and L. Zelnik-Manor, "On sifts and their scales," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, June 2012, pp. 1522–1528.

[25] S. Dreiseitl and L. Ohno-Machado, "Logistic regression and artificial neural network classification models: a methodology review," *Journal of biomedical informatics*, vol. 35, no. 5, pp. 352–359, 2002.

[26] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, USA: Springer, 2006.

[27] R. Battiti, "First- and second-order methods for learning: Between steepest descent and newton's method," *Neural Computation*, vol. 4, no. 2, pp. 141–166, March 1992.

[28] B. Krishnapuram, L. Carin, M. A. Figueiredo, and A. J. Hartemink, "Sparse multinomial logistic regression: Fast algorithms and generalization bounds," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 957–968, 2005.

[29] Y. Huang, R. Wu, Y. Sun, W. Wang, and X. Ding, "Vehicle logo recognition system based on convolutional neural networks with a pretraining strategy," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 1951–1960, Aug 2015.

[30] A. Vedaldi and B. Fulkerson, "VLFeat - an open and portable library of computer vision algorithms," in *Proc. ACM International Conf. on Multimedia*, 2010.

[31] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.

[32] A. A. Lopes, J. R. Bertini Jr, R. Motta, and L. Zhao, "Classification based on the optimal k-associated network," in *Complex Sciences*. Springer, 2009, pp. 1167–1177.

[33] A. Hinneburg, C. C. Aggarwal, and D. A. Keim, "What is the nearest neighbor in high dimensional spaces?" in *Proc. of the 26th International Conf. on Very Large Data Bases*, 2000, pp. 506–515.