A Multi-Feature Fusion Based Vehicle Logo Recognition Approach for Traffic Checkpoint

Sha Ding¹, a, Hongyang Wu², b

¹School of Computing and Engineering Wuhan International Culture University Wuhan, China
²School of Electronic Information Wuhan University Wuhan, China

a474484578@qq.com, bHongyang_wu@hotmail.com

Abstract. In this paper, we propose a vehicle logo recognition algorithm based on multi-feature fusion using a hierarchical classification approach, which can be applied at traffic checkpoints. First, a typical database of vehicle logos is set up based on surveillance images recorded at traffic checkpoints. Next, three features, HOG, Curvature histograms, and GIST, are extracted and three corresponding first level classifiers are trained using the support vector machine (SVM) algorithm. The probability that a certain test sample belongs to a certain kind can be obtained by predicting the sample with each level-1 classifier. All these probabilities are then concatenated and used as features for training a second-level SVM classifier. The resultant new classifier is used for classifying the vehicle logos of the test set. The experimental results show that the proposed approach to hierarchically integrate multiple features provides excellent accuracy for the vehicle logo recognition task.

1. Introduction
Most of the vehicle logo recognition methods, with a few exceptions, are based on features such as invariant moment [1] and techniques such as pattern matching [2]. Feature based approaches can either perform feature matching directly [3] or combine features using different types of classifiers, such as histogram of oriented gradient (HOG) [4] for logo description, support vector machines (SVM) [5] for classification [6], or combination of Haar features [7] and Adaboost classifier [8], combination of SIFT and SVM [9], and combination of extended SIFT and classifiers [10], to achieve state of the art results. In addition, in recent years, convolutional neural networks have been widely adopted for vehicle logo recognition [11]. Since there are always limitations in single feature based approaches for logo description, multi-feature fusion has evolved as a new research trend. Xiao ET. Al [12] successfully used an approach which first extracted three features (color, edge, and descriptor) and then weighted the probabilities acquired from three corresponding classifiers to compute the final classification probability with high accuracy. The above fusion approaches combine results of multiple classifiers using simple methods. In this paper, we propose a two-level hierarchical classification based approach that performs multi-feature fusion using histogram of oriented gradient (HOG) [4], curvature histograms [13], and GIST feature [14] for vehicle logo recognition.
2. Features

To perform vehicle logo recognition, some feature descriptors must be chosen to represent the images. In this paper, the chosen features are HOG, curvature histograms, and GIST.

2.1. HOG Feature

HOG is a feature descriptor which computes a histogram of gradient orientations for different local regions of an image. Since the computation is performed at the level of each cell (small connected regions), this algorithm provides good invariance with respect to geometrical transformation and changes in illumination. The steps involved in HOG feature extraction is shown in Figure 1.

\[
\begin{align*}
\text{input image} & \rightarrow \text{image normalization} & \rightarrow \text{gradient computation} & \rightarrow \text{gradient histogram} & \rightarrow \text{contrast normalization} & \rightarrow \text{HOG vector}
\end{align*}
\]

**Figure 1.** Steps to generate HOG Feature.

First order differential is used to compute the image gradient. Suppose the intensity value at point \((x, y)\) is \(f(x, y)\), the gradient \(\nabla f\) at this point is

\[
\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\delta f(x, y)}{\delta x} \\ \frac{\delta f(x, y)}{\delta y} \end{bmatrix}
\]

(1)

Where \(G_x, G_y\) denote the gradient in the direction of \(X, Y\) respectively. \(\frac{\delta f(x, y)}{\delta x}\) and \(\frac{\delta f(x, y)}{\delta y}\) are first partial differentials, defined as:

\[
\frac{\delta f(x, y)}{\delta x} = f(x+1) - f(x-1)
\]

(2)

For each cell comprising of 2*2 pixels in the image, the histogram of oriented gradient (HOG) based on nine directions is counted to generate the descriptor for the cell. Then, all the cells within one block are serially linked to perform contrast normalization. Thus, the HOG feature is obtained by combining all the blocks of the entire image.

2.2. Curvature histogram Feature

Curvature histogram [13] (henceforth referred to as Curv-H) is a recently introduced feature descriptor. It extends the histogram of orientations to curves, thus allowing us to distinguish local morphology at a tiny level.

Curvature of a curve describes the extent by which it deviates from a straight line. The curvature \(k\) of a point is equal to the inverse of the radius obtained from local curve fitting. The curve \(X\) is parameterized by the arc length \(t\), i.e. it maps the interval \(t\) to an even interval \(X(t)\). Suppose \(T(t)\) is the tangent and \(N(t)\) is the normal line at \(X(t)\), the curvature can be redefined as the change of the normal along the line of tangent:
The faster the normal line rotates along the curve, the higher is the curvature. The trajectory of \( k \) depends on the direction of the normal line. For each pixel \((x, y)\) in the image, we extract the gradient \( g = \left( g_x, g_y \right)^T \) (\( g_x, g_y \) are the gradients in the direction of \( x \) and \( y \) respectively.) using a standard linear difference filter, which provides the tangent vector \( \phi = \left( -g_y, g_x \right)^T \) orthogonal to the gradient. Normalizing both gives us \( g^N = \frac{g}{\|g\|} \) and \( \phi^N = \frac{\phi}{\|\phi\|} \) (\( g^N, \phi^N \) denotes the normalized gradient and tangent vector). The normalized Jacobian determinant (Equation 4) represents the change of the direction of the gradient.

\[
J\left(g^N\right) = \begin{pmatrix}
\left(g^N_x\right)_x & \left(g^N_x\right)_y \\
\left(g^N_y\right)_x & \left(g^N_y\right)_y
\end{pmatrix}
\]

\( J\left(g^N\right) \cdot \phi^N \) Demonstrates the change of gradient along the direction of the tangent vector. From equation 3 we obtain the expected vector as \(-k \cdot T(t)\). However, due to common deviations in discrete images, corrections need to be made by introducing an intermediate scalar variable \( q \), which is expressed as

\[
q = -\left(\phi^N \right)^T \cdot J\left(g^N\right) \cdot \phi^N
\]

The value of \( q \) can be either positive or negative, depending on the convexity/concavity of the curve and the notation of the gradient.

To generate a descriptor, the curvature is combined with the direction of the gradient. The convexity/concavity of the curve can be distinguished by removing the coupling between the notation of the curvature and the sign of the gradient. For this the gradient vector \( g^N \) is multiplied by \( q \), to yield the curvature \( Q \) which is called the “Vector Curvature”:

\[
Q = q \cdot g^N
\]

The vectorized curve for each pixel in the image and the corresponding sparse cluster pixel histogram can hence be calculated. The directions of \( Q \) can be quantified into 8 units. The notation of the direction is based on the notation of curvature instead of the gradient notation. Finally, the curvature histogram feature descriptor can be obtained by a combination of the direction, notation and range of the curvature.

2.3. GIST Descriptor
The GIST feature is based on the same contextual information which the human eyes rely on for image recognition. It starts by filtering the image using a group of Gabor filters to extract the contour information of the filtered image. A two dimensional Gabor function is shown as follows:
\( g(x, y) = \frac{1}{2\pi \sigma_x \sigma_y} \exp \left[ -\left( \frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2} \right) \right] \cos(2\pi f_0 + \phi) \)  
\( (7) \)

Where \( (x, y) \) is the pixel coordinate; \( f_0 \) is the frequency; \( \sigma_x \) and \( \sigma_y \) are the Gaussian variances in the directions of \( x \) and \( y \); and \( \phi \) is the phase difference of the cosine harmonic wave factor. By scaling and rotating a 2D Gabor filter, a set of multi-scale Gabor filters can be established as:

\[
\begin{align*}
& g_{mn}(x, y) = a^{-m} g'(x', y') \\
& x' = a^{-m} (x \sin \theta + y \cos \theta) \\
& y' = a^{-m} (-x \sin \theta + y \cos \theta) \\
& \theta = n\pi / (n+1)
\end{align*}
\]  
\( (8) \)

Where \( \theta \) defines the direction of the filter; \( a \) is the expansion factor; \( m \) is the scale number; and \( n \) is the direction number. Different value of \( m \) and \( n \) correspond to filters with different scales.

The generation process of GIST feature can be defined as follows: For an input grayscale image with the size of \( \text{h} \times \text{w} \), divide it into \( n_s \times n_s \) grids of identical size. Each grid has the size of \( \text{h} / n_s \times \text{w} / n_s \), where \( h = \frac{\text{h}}{n_s} \) and \( w = \frac{\text{w}}{n_s} \). Now filter each grid with the filter set with \( n_c \) paths. The feature is obtained by serially linking all the filtered results. The feature of the \( i \)-th grid is

\[
\text{GIST}_i(x, y) = \text{cat}(f_i(x, y) \ast g_{mn}(x, y))
\]  
\( (9) \)

Where \( i = 1, 2, 3, \ldots, n_s \times n_s \); \( f_i(x, y) \) is the image of the \( i \)-th grid; and \( \text{cat}() \) means the concatenation of vectors. The number of dimensions of \( \text{GIST}_i \) is \( h_0 \times w_0 \times n_c \). The average value of the features from all the grids is used as the GIST feature, i.e

\[
\overline{\text{GIST}_i} = \frac{1}{h_0 \times w_0} \sum_{(x,y)} \text{GIST}_i(x, y)
\]  
\( (10) \)

Where \( \overline{\text{GIST}_i} \) is the mean feature value after path \( i \). The concatenation of every mean feature value is the GIST feature for the whole image and has \( n_s \times n_s \times n_c \) dimensions.

3. Multi-Feature Fusion Algorithm

This paper proposes a two-level hierarchical classifier for multi-feature fusion. The detailed steps are shown in Figure 2.
The three features in the figure are HOG, GIST feature, and Curvature histogram (Curv-H). For our classification task, we choose a certain proportion of samples as the training set for the level-1 classifier. The rest are used as the test set for the level-1 classifier. An SVM classifier is employed to train the system and the level-1 test samples are predicted to obtain the soft probabilities corresponding to the three features. Next we concatenate the three soft probabilities and use the concatenated soft probabilities as a feature. Since only level-1 test samples have soft probabilities, the proportion of data selected as the level two training sample is the same as the proportion selected for the level-1 classifier. The rest can be used as level-2 test samples. SVM is used to train the level-2 classifier. Then this second level trained SVM model is applied to make the final prediction for the category of vehicle logos on the level-2 test set, demonstrating the effectiveness of a two-level SVM classifier.

The purpose of the fusion approach is to perform concatenation of the soft probabilities. When predicting test samples, the classifier based on the $j$th feature gives the result $P_j$, which is the probability distribution that a sample belongs to each class, i.e. $P_j = \{p_{j,1}, p_{j,2}, \ldots, p_{j,N}\}$ ($p_{j,i}$ denotes the probability that a sample belongs to class $i$ where the number of classes is $N$). Each sample will eventually be classified as the class with the highest probability. Suppose $J$ features are used for training to obtain $J$ classifiers, then for each sample, there will be $J$ soft probabilities. Since this paper uses concatenation for fusion, the soft probability after fusion can be written as:

$$P = \{p_{1,1}, p_{1,2}, \ldots, p_{1,N}, \ldots, p_{J,1}, \ldots, p_{J,N}\}$$

All the three bottom layer features of the images can be used for classification. However, using a single feature is usually not sufficient for multi-class image classification. Complementary relationships may exist among different features used to represent the image. In such cases, adaptive fusion of multiple features can lead to better accuracy. Thus, this paper proposes a two-level hierarchical classification model based on the deep fusion of multi-features. Suppose the three features, HOG, Curv-H, and GIST, extracted from an image are $f_1$, $f_2$, and $f_3$ respectively and the three soft probabilities estimated by the level-1 SVM classifier are $p_1$, $p_2$, and $p_3$. A vector can be used to represent the probability distribution for the category of the image, where the length of the vector is the number of classes. We use equation 13 to concatenate the three features:

$$Y_s = cat(p_1, p_2, p_3) \in \mathbb{R}^{3C}$$

Figure 2. Multi-Feature Fusion Process.
After the fusion of the soft probabilities, we use the same type of classifier (SVM) for training, to ensure the consistency of the data. In order to verify the complementary relationship between the features, we use different combinations of HOG, Curv-H, and GIST features. In particular, we use the combination a) HOG and Curv-H, b) HOG and GIST, c) Curv-H and GIST, and d) all three together at the test stage and compare the results.

4. Experiments and Analysis

4.1. Vehicle Logo Database

In order to test and analyze the algorithm proposed in this paper, we established a vehicle logo database. Images used in the experiments are recorded by surveillance cameras at traffic checkpoints. Logos from eight common car brands, as shown in Figure 3, are selected. These brands are: Honda, Peugeot, BYD, VW, Toyota, Kia, Hyundai, and Citroen.

![Figure 3. Different Vehicle Logos](image)

The distribution of the number of vehicles for each brand in the database is shown in Table 1.

| Vehicle brand | Honda | Peugeot | BYD | VW | Toyota | Kia | Hyundai | Citroen |
|---------------|-------|---------|-----|----|--------|-----|---------|---------|
| Vehicle number| 135   | 85      | 132 | 129| 158    | 144 | 143     | 164     |

To better simulate a real-world scenario to test the accuracy of the program, the images selected for the experiments were taken both at day time and at night. The image quality and clarity of the chosen images vary; and the distance of the images from the check points vary as well.
4.2. Vehicle Logo Classification
To evaluate the performance of the algorithm, all the following experiments are executed when the logos are successfully identified from the images. We use the RBF kernel of the libsvm library to train the classifier. To ensure the fairness of the test results, we randomly selected 5%, 15%, 30%, and 50% of the samples from the 8 brands as training data and use the rest as test data. For each sample data proportion, we repeat the experiments 5 times and report the average value as the final result. When representing the model at the middle layer of the experiments, SIFT features are extracted as features from the bottom layer. After repeated tests, the sliding window for the extraction of the SIFT feature is determined to be 8*8 and the step length is set as 2. The maximum accuracy is obtained when the dictionary size is selected to be 50. All images are normalized to the size of 32*32. Samples from 8 brands are used in this paper, i.e. $N = 33$. Three features are used, i.e. $J = 3$. Thus, the feature vector of the soft probabilities after fusion has the length of $33 \times 3 = 99$.

4.3. Experimental Results and Analysis
To evaluate the performance of the multi-feature fusion algorithm, we compare the classification results of a single feature with multi-feature fusion, as shown in Table 2 and Table 3.

Table 2. Classification results with a single feature

| Training sample proportion | 5%     | 15%     | 30%     | 50%     |
|----------------------------|--------|---------|---------|---------|
| HOG                        | 77.29% | 85.71%  | 93.03%  | 95.23%  |
| Curv-H                     | 88.53% | 90.65%  | 94.48%  | 94.10%  |
| GIST                       | 85.54% | 91.32%  | 95.13%  | 97.27%  |

Table 3. Classification results with multi-feature fusion

| Training sample proportion | 5%     | 15%     | 30%     | 50%     |
|----------------------------|--------|---------|---------|---------|
| HOG+Curv-H                 | 85.78% | 92.52%  | 96.92%  | 99.09%  |
| HOG+GIST                   | 86.14% | 95.33%  | 98.22%  | 99.55%  |
| Curv-H+GIST                | 91.04% | 93.99%  | 97.73%  | 97.05%  |
| HOG+Curv-H+GIST            | 89.84% | 94.52%  | 99.03%  | 99.77%  |

It can be observed from the results that the classification accuracy is generally high when the proportion of training samples is larger than 30%. When the proportion of training samples is over 50%, the classification accuracy is higher than 94%.

Comparing the results with different proportions of training samples show that increasing the amount of training samples increased the accuracy of the classification. It can also be noted from the tables that feature fusion performs better than using a single feature, especially when the number of training samples is low. Feature fusion with a small number of training samples can achieve classification results which are equivalent to the results obtained with a large number of training samples for a single feature, suggesting that using multiple features to describe samples can compensate for the limitation in accuracy for small training sample sizes.

5. Conclusion
In this paper, we extend the idea of a simple combination of the classification results from multiple classifiers and propose a two-level hierarchical classifier based multi-feature fusion approach. This approach concatenates the soft probabilities obtained from the level-1 classifiers that use different features and then uses these concatenated probabilities as input features for training the second level classifier. Then this second level classifier is used to predict the classification probability of samples to classify vehicle logo brands. After comparing the experimental results using different proportions of training samples, we notice that multi-feature hierarchical fusion can lead to excellent classification results even for lower number of training samples.
References

[1] Soon F C, Khaw H Y, Chuah J H. Pattern recognition of Vehicle Logo using Tchebichef and Legendre moment [C]//2015 IEEE Student Conference on Research and Development (SCoReD). IEEE, 2015: 82 - 86.

[2] Wang Y, Liu Z, Xiao F. A fast coarse-to-fine vehicle logo detection and recognition method[C]//Robotics and Biomimetics, 2007. ROBIO 2007. IEEE International Conference on. IEEE, 2007: 691 - 696.

[3] Psyllos A P, Anagnostopoulos C N E, Kayafas E. Vehicle logo recognition using a sift-based enhanced matching scheme [J]. IEEE transactions on intelligent transportation systems, 2010, 11 (2): 322 - 328.

[4] Dalal N, Triggs B. Histograms of oriented gradients for human detection[C]/2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). IEEE, 2005, 1: 886 - 893.

[5] Vapnik V. The nature of statistical learning theory[M]. Springer Science & Business Media, 2013.

[6] Llorca D F, Arroyo R, Sotelo M A. Vehicle logo recognition in traffic images using HOG features and SVM[C]//16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013). IEEE, 2013: 2229 - 2234.

[7] Lienhart R, Maydt J. An extended set of haar-like features for rapid object detection [C]//Image Processing. 2002. Proceedings. 2002 International Conference on. IEEE, 2002, 1: I-900-I-903 vol. 1.

[8] Sam K T, Tian X L. Vehicle logo recognition using modest adaboost and radial tchebichef moments [C]//International Conference on Machine Learning and Computing (ICMLC 2012). 2012.

[9] Ou Y, Zheng H, Chen S, et al. Vehicle logo recognition based on a weighted spatial pyramid framework [C]//17th International IEEE Conference on Intelligent Transportation Systems (ITSC). IEEE, 2014: 1238 - 1244.

[10] Yu S, Zheng S, Yang H, et al. Vehicle logo recognition based on bag-of-words [C]//Advanced Video and Signal Based Surveillance (AVSS), 2013 10th IEEE International Conference on. IEEE, 2013: 353 - 358.

[11] Huang Y, Wu R, Sun Y, et al. Vehicle Logo Recognition System Based on Convolutional Neural Networks With a Pretraining Strategy [J]. IEEE Transactions on Intelligent Transportation Systems, 2015, 16 (4): 1951 - 1960.

[12] Xiao G R, Dong Y, Liu Z X, et al. Supervised TV logo Detection Based on SVMs [C]. 2010 2nd IEEE International Conference on Network Infrastructure and Digital Content (IC-NIDC 2010). New York: IEEE Press.2010: 174 - 178

[13] Fischer P, Brox T. Image Descriptors Based on Curvature Histograms [M]. New York: Pattern Recognition. Springer International Publishing, 2014: 239 - 249.

[14] Oliva A, Torralba A. Modeling the Shape of the Scene: a Holistic Representation of the Spatial Envelope [J]. International journal of computer vision, 2001, 42(3): 145 - 175.