Multi-class Vehicle Type Recognition System

Xavier Clady[1], Pablo Negri[1], Maurice Milgram[1], and Raphael Poulenard[2]

[1] Université Pierre et Marie Curie-Paris 6, CNRS FRE 2907
Institut des Systèmes Intelligents et Robotique
[2] LPR Editor - Montpellier
xavier.clady@upmc.fr

Abstract. This paper presents a framework for multiclass vehicle type (Make and Model) identification based on oriented contour points. A method to construct a model from several frontal vehicle images is presented. Employing this model, three voting algorithms and a distance error allows to measure the similarity between an input instance and the data bases classes. These scores could be combined to design a discriminant function. We present too a second classification stage that employ scores like vectors. A nearest-neighbor algorithm is used to determine the vehicle type. This method have been tested on a realistic data set (830 images containing 50 different vehicle classes) obtaining similar results for equivalent recognition frameworks with different features selections [12]. The system also shows to be robust to partial occlusions.

1 Introduction

Many vision based Intelligent Transport Systems are dedicated to detect, track or recognize vehicles in image sequences. Three main applications can be distinguished. Firstly, embedded cameras allow to detect obstacles and to compute distances from the equiped vehicle [15]. Secondly, road monitoring measures traffic flow [2], notifies the health services in case of an accident or informes the police in case of a driving fault. Finally, Vehicle based access control systems for buildings or outdoor sites have to authentify incoming (or outcoming) cars [12]. The first application has to classify region-of-interest (ROI) in two classes: vehicles or background. Vehicles are localized in an image with 2D or 3D bounding box [10,15]. The second one can use geometric models in addition to classify vehicles in some categories such sedans, minivans or SUV. These 2D or 3D geometric models are defined by deformable or parametric vehicle templates [5,6,7].

Rather than these two systems, the third one uses often only the recognition of a small part of vehicle : the license plate. It is enough to identify a vehicle, but in practice the vision based number plate recognition system can provide a wrong information, due to a poor image quality or a fake plate. Combining such systems with others process dedicated to identify vehicle type (brand and model) the authentication can be increased in robustness (see fig. 1). This paper adresses the identification problem of a vehicle type from a vehicle greyscale frontal image: the input of the system is an unknown vehicle class, that the system has to determine from a data base.

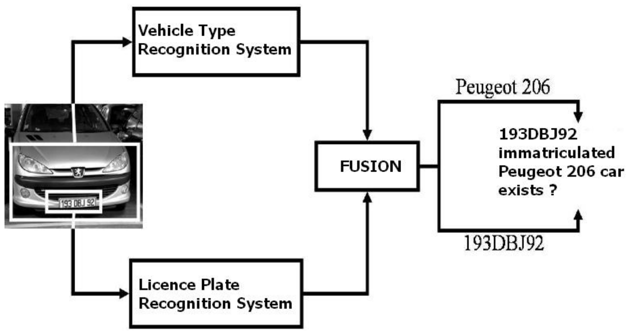

Vehicle Type Recognition System
Licence Plate Recognition System
FUSION
Peugeot 206
193DBJ92 immatriculated Peugeot 206 car exists ?
193DBJ92

Fig. 1. The fusion system

Few papers deal with a similar problem. In a recognition framework for rigid object recognition, Petrovic and Cootes [12] tested various features for vehicle type classification. Their decision module is based on two distance measures (with or without Principal Component Analysis pre-stage): the dot product $d = 1 - f_1 f_2$ and the Euclidean measure $d = |f_1 - f_2|$, where f_i is the feature vectors. The dot product gives slighthly outperforming results. Best results are obtained with gradients based representations. These results can be explained because the vehicle rigid structure is standardized by the manufacturer for each model. The relevant information contained in contour edge and orientation is independent of the vehicle color. Daniel T.Munroe et al [14] studied machine learning classification techniques applied on features vectors (extracted with a Canny edge detector). L. Dlagnekov [3] used Scale Invariant Feature Transforms (SIFT) to compute and match keypoints. Zafar et al. [18] used a similar algorithm. David A. Torres[16] extended the work of Dlagnekov by replacing the SIFT features with features which characterize contour lines. In [9], Kazemi et al investigated use of Fast Fourier Transforms, Discrete Wavelet Transforms and Discrete Curvelet Transforms based image features. All these works used gradient or contour based features.

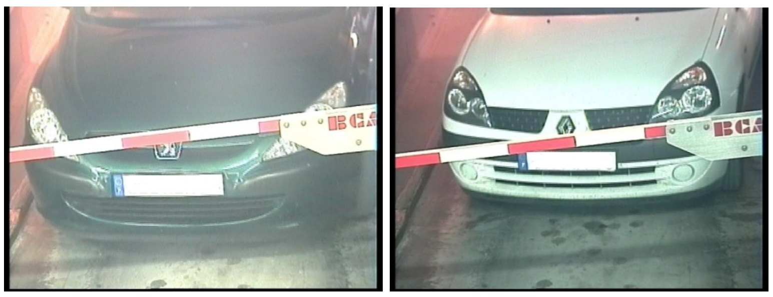


Fig. 2. Real vehicle images with the tollgate presence

In this paper, a multiclass recognition system is developed using the oriented-contour pixels to represent each vehicle class. The system analyses a vehicle frontal view identifying the instance as the most similar model class in the data base. The classification is based on a voting process and a Euclidean edge distance. The algorithm have to deal with partial occlusions. Tollgates hide a part of the vehicle (see fig. 2) and making inadequate the appearance-based methods. In spite of tollgate presence, our system doesn't have to change the training base or apply time-consuming reconstruction process.

In section 2, we explain how we define a model for every class in the data base using the oriented-contour points. Section 3 employs this model to obtain *scores* measuring the similarity between the input instance and the data bases classes. These scores could be combined to design a discriminant function. We present too a second classification stage that employ scores like vectors. A nearest-neighbor algorithm is used to classify the vehicle type. Results of our system are presented in the section 4. We finish with conclusions and perspectives.

2 Model Creation

During the initial phase of our algorithm, we produce a model for all the K vehicle types classes composing the system knowledge. The list of classes the system is capable to recognize is called Knowledge Base (**KnB**). In our system, the Knowledge Base will be the **50** vehicle type classes.

2.1 Images Databases

All ours experiments have been carried out on the Training Base (**TrB**) and on the Test Base (**TsB**). The **TrB** samples (291 images) are used to produce the oriented-contour point models of the vehicle classes. While the **TsB** samples (830 images) are utilized to evaluate the performance of the classification system. In figure 3, the upper row shows samples from **TrB** and in the bottom row, the figure shows the corresponding vehicle type class of the **TsB**. These databases are composed of frontal vehicle views, captured in different car parks, under different light conditions and different points of view.

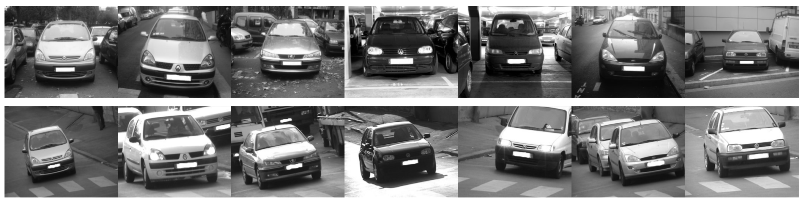


Fig. 3. In the upper row, the figure shows samples from **TrB**. In the bottom row, the figure shows the corresponding vehicle type class of the **TsB**.

2.2 Prototype Image

We create a canonical rear-viewed vehicle image I from the four corner points of the license plate $\{\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{D}\}$ (see fig. 4). The image templates are called prototypes and in the present work are 600 * 252 pixels (rows * columns). A ROI defined by the points $\{\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{D}\}$ is independent of the vehicle location in the image and the scale (fig.4.a). In order to correct the orientation of the original image (see example in fig.3), an affine transformation moves original points $\{\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{D}\}$ to the desired $\{\mathbf{A'},\mathbf{B'},\mathbf{C'},\mathbf{D'}\}$ reference position, considering the vehicle grille and the license plate in the same plane. A license plate recognition system provides the corners of the vehicle license plate.

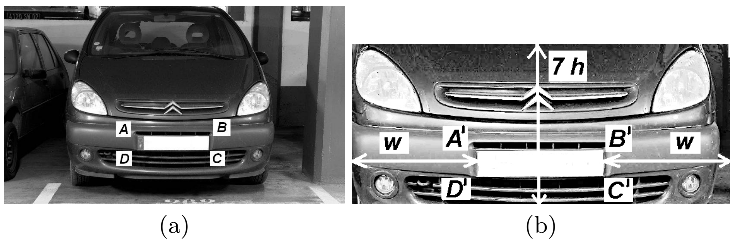


Fig. 4. (a) original image, (b) prototype I

The Sobel operator is used to compute the gradient's magnitude and orientation of the greyscale prototype I $(|\nabla g_I|, \phi_I)$. An oriented-contours points matrix $\mathbf{E}_I$ is obtained using an histogram based threshold process. Each edge point $\mathbf{p}_i$ of $\mathbf{E}_I$ is considered as a vector in $\Re^3$: $\mathbf{p}_i$=[x,y,o]', where (x,y) is the point position, and o is the gradient orientation of $\mathbf{p}_i$ [11]. We sample the gradient orientations to N bins. To manage the cases of vehicles of the same type but with different colors, the modulus π is used instead of the modulus 2π [1]. In the present application, $N = 4$.

2.3 Model Features

Oriented-Contour points features array. Each class in the **KnB** is represented by n prototypes in the **TrB**. This quantity n varies from class to class, having some defined with one prototype only.

Superposing the n prototypes of the class k, we find an array of the redundant oriented-contour points. This feature array of Oriented-Contour based points models this class in the **KnB**. The algorithm operates the n prototypes of the class k in the **TrB** by couples (having $C_{n,2}$ couples at all). Let be $(\mathbf{E}_i, \mathbf{E}_j)$ a couple of Oriented-Contour Points matrix of the prototypes 1 and 2 from the k class. We define an 600x252xN accumulator matrix A_{ij} and the vote process is as follow: a) taking a point $\mathbf{p}_i$ of $\mathbf{E}_i$, we seek in $\mathbf{E}_j$ the nearest point $\mathbf{p}_j$ with the same gradient orientation; b) the algorithm increments the accumulator A_{ij}

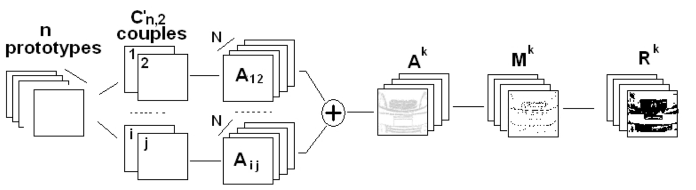


Fig. 5. Model creation

in the middle point of $\overline{\mathbf{p}_i\mathbf{p}_j}$ at the same gradient orientation; c) the procedure is repeated for all the points $\mathbf{p}_i$ of $\mathbf{E}_i$. Considering the addition of all A_{ij} we obtain the accumulator array A^k: $A^k = \sum_{i,j} A_{ij}$. The most voted points $\mathbf{a}_m$=$[x$,y,$o]$ of A^k are selected iteratively. We impose a distance of 5 pixels between the $\mathbf{a}_m$ in order to obtain a homogeneous distribution of the model points. We store $\mathbf{a}_m$ in a feature array $\mathbf{M}^k$. The array $\mathbf{M}^k$ contains the Oriented-Contour Points that are rather stable through the n samples of the class k.

When $n = 1$, the accumulator matrix A^k cannot be computed: the feature array $\mathbf{M}^k$ is then determined from the maximum values of the gradient magnitude $|\nabla g_I|$.

Weighted Matrix. The Chamfer distance is applied to determine the distance from every picture element to the given $\mathbf{M}^k$ set (fig. 6). This figure shows the four R_i^k Chamfer region matrix (one for each gradient orientation) obtained after threshold the Chamfer chart matrix D_i^k with the distances smaller than r.

Two weighted regions arrays W_+^k and W_-^k will be created for each class k. W_+^k is based on the R^k region matrix where each pixel point has a weight related to the discrimination power of the corresponding oriented-contour points. Pixel

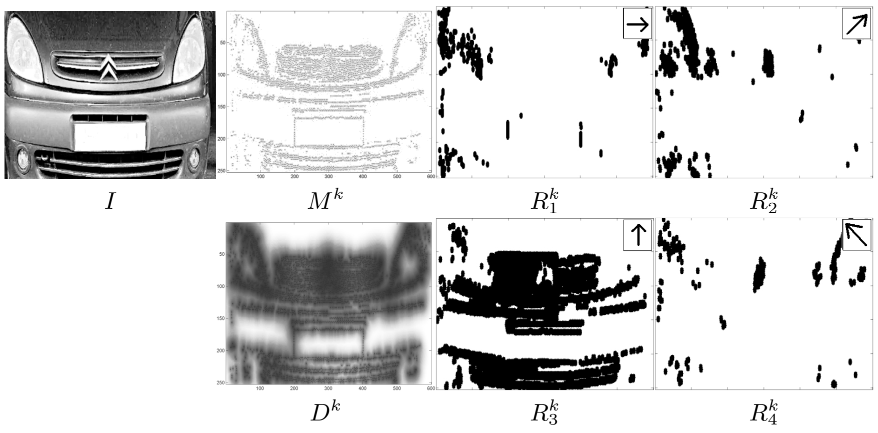


Fig. 6. Obtaining Chamfer region matrix

points rarely present in the others classes obtain highest weights. We give lowest weights to the points present in the majority of the Knowledge Base classes.

$$W_+^k = \frac{1}{K-1} \sum_{i,i \neq k} (R^k - R^i \cap R^k)$$

W_-^k gives a negative weight to the points of the other models which are not present in the matrix R^k of the model k. Pixel points that are present in most of the other classes obtain highest weight values. In the other hand, pixel points present in few classes get lowest weight values.

$$W_-^k = -\frac{1}{K-1} \sum_{i,i \neq k} (R^i - R^i \cap R^k)$$

The K classes in the **KnB** are modelled by $\{\mathcal{M}_1, ..., \mathcal{M}_K\}$, where each $\mathcal{M}_k = \{\mathbf{M}^k, W_+^k, W_-^k\}$.

3 Classification

This section develops the methods to classify the samples providing from the $\mathcal{T}\mathit{\int}\mathcal{B}$ using the models $\mathcal{M}_k$. A new instance t is evaluated on the classification function $G(t) = ArgMax\{g_1(t), ..., g_K(t)\}$ using the *winner-take-all* rule. The example t is labelled by $k \in \mathcal{K}$ from the highest score of the g_k. Two types of matching scores compose the g_k (see fig. 7). The first obtains a score based on three kind of votes (positive, negative and class votes) for each class k. The second score evaluates the distance between the oriented-contour points of the model $\mathbf{M}^k$ to the oriented-contour points of t.

Obtaining the image prototype of the sample t from the Test Base, we calculate the oriented-contour points matrix $\mathbf{E}_t$ (section 2.2). Considering the large number of points in $\mathbf{E}_t$, we have to choose a limited set of T points. The value

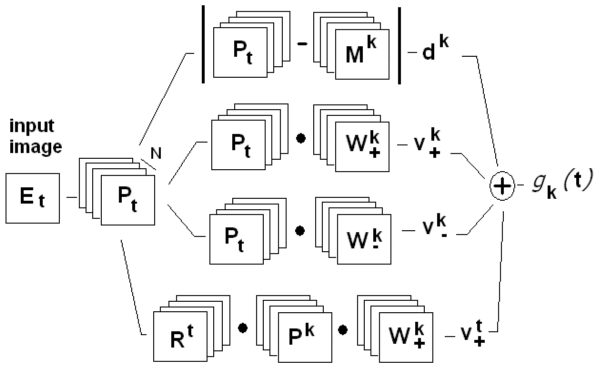


Fig. 7. Obtaining the discriminant function

of T is a compromise between the computing time and a good rate of correct classifications (in our algorithm, $T = 3500$). To select these points, we construct a sorted list of the prototype positions (x, y, o). We sort in decreasing order, the values of the weighted arrays W_+^i $i = 1, ..., K$, placing the discriminant pixels (highest values) in the firsts positions of the list. Looking iteratively if the pixels in the list are present in $\mathbf{E}_t$, we pick up the T points, and place them in $\mathbf{P}_t$.

3.1 Designing the Discriminant Function

Positive votes. The methodology consists in accumulating votes for the class k, whenever a point of $\mathbf{P}_t$ falls in a neighbourhood of a $\mathbf{M}^k$ point. We define the neighbourhood of the point $\mathbf{M}^k$ as a circle of radius r around the point of interest. This neighbourhood representation is modelled in the Chamfer regions R_i^k. Moreover, each point of $\mathbf{P}_t$ votes for the class k with a different weight depending on its value in the matrix W_+^k.

The nonzero points of the dot product of $\mathbf{P}_t$ and W_+^k correspond to the points of $\mathbf{P}_t$, that belong to a neighbourhood of the $\mathbf{M}^k$'s points. Thereafter, we calculate the amount of positive votes in equation 1 where [$\bullet$] is the dot product.

$$v_+^k = \sum_x \sum_y \sum_o \mathbf{P}_t \bullet W_+^k \tag{1}$$

Negative votes. The negative votes take into account the points of $\mathbf{P}_t$ that did not fall into the neighbourhood of the $\mathbf{M}_k$ points. We punish the class k by accumulating these points weighted by the matrix W_-^k. The amount of negative votes is defined as:

$$v_-^k = \sum_x \sum_y \sum_o \mathbf{P}_t \bullet W_-^k$$

Votes to test. We calculate the votes from the models to the sample test. In short, the method is the same as the one detailed in the preceding section. We first build the chart of Chamfer Distances for $\mathbf{E}_t$. We keep the regions around the oriented-contour points of $\mathbf{E}_t$ which are at a distance lower than r pixels in the matrix R^t. Then, randomly selecting T points from the array $\mathbf{M}_k$, we obtain a representation of this set in an array $\mathbf{P}^k$. Each point of the matrix $\mathbf{P}^k$ is weighted by the matrix W_+^k. Total votes from the class k to the sample test t are calculated as:

$$v_+^t = \sum_x \sum_y \sum_o R^t \bullet \mathbf{P}^k \bullet W_+^k$$

Distance Error. The last score is the error measure of matching the $\mathbf{P}_t$ points with their nearest point in $\mathbf{M}_k$. Calculating the average of all the minimal distances, we obtain the error distance d^k [4]:

$$H(\mathbf{P}_t, \mathbf{M}_k) = max(h(\mathbf{P}_t, \mathbf{M}_k), h(\mathbf{M}_k, \mathbf{P}_t))$$

with :

$$h(\mathbf{P}_t, \mathbf{M}_k) = mean_{a \in \mathbf{P}_t}(min_{b \in \mathbf{M}_k} \|a - b\|))$$

Furthermore, values in the error vector have to be processed by a decreasing function considering that in the vote vectors we search for the maximum and for the error vector we search for the minimum.

3.2 Classification Strategies

We have developed two strategies for classification. The first combines the scores in a discriminant function. The second creates voting vector spaces from the scores : the decision is based on a nearest-neighbor process.

First Strategy : Discriminant Function. The four matching scores $\{v_+^k, v_-^k, v_+^t, d^k\}$ are combined in a discriminant function $g_k(t)$ matching the sample test t to the class k. A pseudo-distance of Mahalanobis normalizes the scores: $\bar{v} = (v - \mu)/\sigma$, where (μ, σ) are the mean and the standard deviation of v. The discriminant function is defined as a fusion of scores:

$$g_k(t) = \alpha_1 \ \bar{v}_+^k + \alpha_2 \ \bar{v}_-^k + \alpha_3 \ \bar{v}_+^k + \alpha_4 \ \bar{d}^k \tag{2}$$

The α_i are coefficients which weight each classifier. In our system, we give the same value for all α_i.

Finally, given the test sample t, its class label k is determined from:

$$k = G(t) = ArgMax\{g_1(t), ...g_K(t)\}$$

Second Strategy : Voting Spaces. We construct vector spaces with the results from the voting process. We define:

- $v(t) = \left[v_{+k}^{mh}, v_{-k}^{mh}, v_{+t}^{mh}, d_k^{mh}\right]_{k=1..K}$ as a vector in a 200(=4 $*$ K) dimension space, called Ω_{wf} (wf = without fusion).
- $v^{PCAX}(t) = \left[v_{+k}^{mh}, v_{-k}^{mh}, v_{+t}^{mh}, d_k^{mh}\right]_{k=1..K}^{PCAX}$ as a vector in a X dimension space, called Ω_{wf}^{PCAX}, (with a Principal Component Analysis pre-stage).
- $g(t) = [g_k(t)]_{k=1..K}$ as a vector in a 50 (=K) dimension space, called Ω_f (f = with fusion).

In these spaces, given the test sample t, its class label is determined as the nearest-neighbor class. It needs reference samples. We use a cross-validation process : the test database is decomposed in two equal parts. The first is used as references. The second is used for the test.

4 Results

With the first strategy, the system correctly identifies 80,2% of 830 test samples. The mean of the recognition rates per class is 69,4%.

The second strategy obtains better results (in mean, with 100 randomly different repartitions):

- in the first space, Ω_{wf}, we obtain 93,1% for the correctly identification rate (83,5% for the mean of the recognition rates per class).

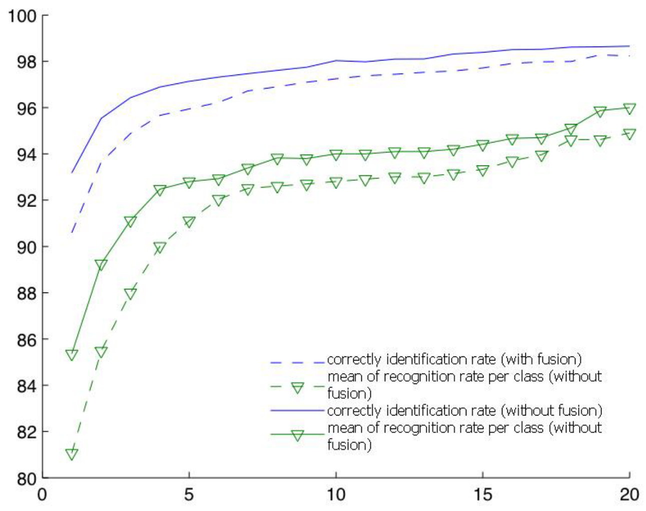


Fig. 8. CMC curves in the Ω_{wf} space (solid line) and in the Ω_f space (dashed line)

- in a second space, Ω_{wf}^{PCA50}, we obtain 86,2% for the correctly identification rate (78,8% for the mean of the recognition rates per class).
- in the last space Ω_f, we obtain 90,6% for the correctly identification rate (86,4% for the mean of the recognition rates per class).

The figure 8 shows the *Cumulative Match Characteristic* curves (CMC[1]). We clearly see that the second strategy in the first space (without fusion and without PCA) gives better results, but with a higher computational cost (due to a high dimensional space). The figure 9 shows us that, without fusion, we have to keep a space dimension higher than 100. Furthermore, a better algorithm performance could be obtained by choosing optimized values for the α_i in the equation2[2]. Moreover, the recognition rate depends on the used reference samples proportion (see figure 10).

Another test simulates the presence of a tollgate at four different locations; in a car park access control system it is difficult to define the relative vertical position between the barrier and the vehicle even if the license plate is always visible. The results for each tollgate position are showed in figure 11. The better recognition are obtained if the virtual tollgate hides the upper part of the images: a lot of noise points are extracted from this part (see figure 13). These points

[1] A Cumulative Match Characteristic (CMC) curve plots the probability of identification against the returned 1:N candidate list size. It shows the probability that a given user appears in different sized candidate lists. The faster the CMC curve approaches 1, indicating that the user always appears in the candidate list of specified size, the better the matching algorithm.

[2] A training algorithm method could be used, but we have to capture more frontal view vehicle samples.

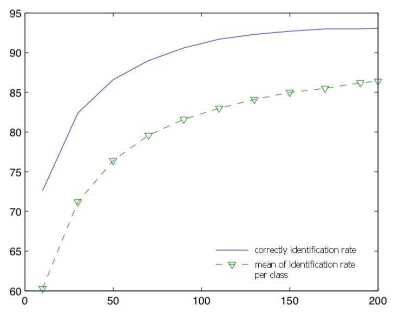


Fig. 9. Recognition rates related to the X dimension in the Ω_{wf}^{PCAX}

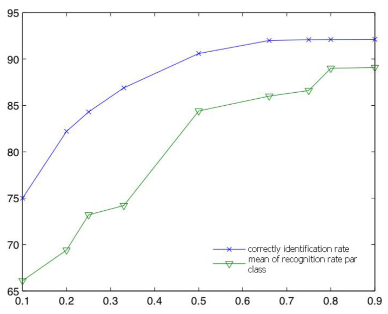
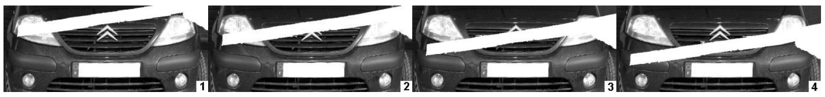


Fig. 10. Recognition rates related to the samples proportion used as reference, in Ω_f



Virtual tollgate position	First Stategy	Ω_{wf} space	Ω_{wf}^{PCA50} space	Ω_f space
1	84,0 %	87,3 %	87,1 %	89,0 %
2	78,5 %	84,5 %	84,1 %	85,6 %
3	78,6 %	84,5 %	83,8 %	85,3 %
4	80,2 %	87,5 %	85,9 %	87,4 %

Fig. 11. The four positions of a virtual tollgate and the recognition rates

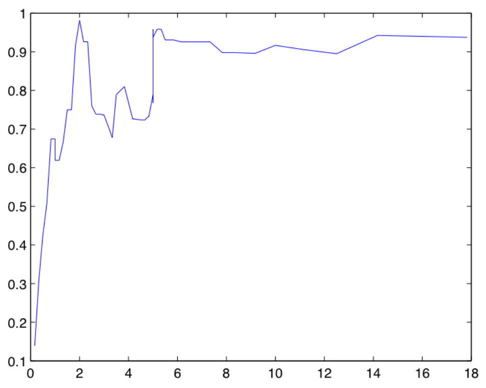


Fig. 12. Mean of the recognition rates par class, related to the images number used in the model creation.

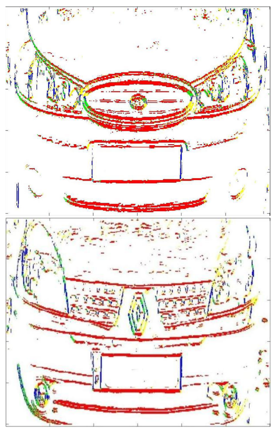


Fig. 13. Results of the contour extraction process

perturb the recognition system. They are filtered if the number of images used in the model creation is sufficient (> 5) as we can see in the figure 12.

5 Conclusions

This article has presented a voting system for the multiclass vehicle type recognition based on Oriented-Contour Points Set. Each vehicle class is composed from one or many grayscale frontal images of one vehicle type (make and model). A discriminant function combines the scores provided from three voting based classifiers and an error distance. A second strategy consists in considering the scores as elements of a vector. A nearest-neighbor process is used to determine the vehicle type. We have tested this method on a realistic data sets of 830 frontal images of cars. The results showed that the method is robust to a partial occlusion of the patterns. The second strategy obtains better results, particularly if the fusion scores are used. Our recognition rate is over 90%. Without occlusion, our system obtains similar results as others works [12]. Future works will be oriented to reduce the influence of the images number used in the model creation process : it could be interesting to recognize the type of a vehicle with only one reference image per class.

References

1. Cootes, T., Taylor, C.: On representing edge structure for model matching. In: Conference on Vision and Pattern Recognition, Hawai, USA, December 2001, vol. 1, pp. 1114–1119 (2001)
2. Douret, J., Benosman, R.: A multi-cameras 3d volumetric method for outdoor scenes: a road traffic monitoring application. In: International Conference on Pattern Recognition, pp. III: 334–337 (2004)
3. Dlagnekov, L.: Video-based car surveillance: License plate make and model recognition. Masters Thesis, University of California at San Diego (2005)
4. Dubuisson, M., Jain, A.: A modified hausdorff distance for object matching. In: International Conference on Pattern Recognition, vol. A, pp. 566–569 (1994)
5. Dubuisson-Jolly, M., Lakshmanan, S., Jain, A.: Vehicle segmentation and classification using deformable templates. IEEE Transactions on Pattern Analysis and Machine Intelligence 18(3), 293–308 (1996)
6. Ferryman, J.M., Worrall, A.D., Sullivan, G.D., Baker, K.D.: A generic deformable model for vehicle recognition. In: British Machine Vision Conference, pp. 127–136 (1995)
7. Han, D., Leotta, M.J., Cooper, D.B., Mundy, J.L.: Vehicle class recognition from video-based on 3d curve probes. In: VS-PETS, pp. 285–292 (2005)
8. Hond, D., Spacek, L.: Distinctive descriptions for face processing. In: British Machine Vision Conference, University of Essex, UK (1997)
9. Kazemi, F.M., Samadi, S., Pooreza, H.R., Akbarzadeh-T, M.R.: Vehicle Recognition Based on Fourier, Wavelets and Curvelet Transforms - a Comparative Study. In: IEEE International conference on Information Technology, ITNG 2007 (2007)
10. Lai, A.H.S., Fung, G.S.K., Yung, N.H.C.: Vehicle type classification from visual-based dimension estimation. In: IEEE International System Conference, pp. 201–206 (2001)

11. Olson, C.F., Huttenlocher, D.P.: Automatic target recognition by matching oriented edge pixels. IEEE Transactions on Image Processing 6(1), 103–113 (1997)
12. Petrovic, V.S., Cootes, T.F.: Analysis of features for rigid structure vehicle type recognition. In: British Machine Vision Conference, vol. 2, pp. 587–596 (2004)
13. Petrovic, V.S., Cootes, T.F.: Vehicle Type Recognition with Match Refinement. In: International Conference on Pattern Recogntion, vol. 3, pp. 95–98 (2004)
14. Munroe, D.T., Madden, M.G.: Multi-Class and Single-Class Classification Approaches to Vehicle Model Recognition from Images. In: AICS (2005)
15. Sun, Z., Bebis, G., Miller, R.: On-road vehicle detection: A review. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(5), 694–711 (2006)
16. Torres, D.A.: More Local Structure for Make-Model Recognition (2007)
17. Zafar, I., Acar, B.S., Edirisinghe, E.A.: Vehicle Make and Model Identification using Scale Invariant Transforms. In: IASTED (2007)
18. Zafar, I., Edirisinghe, E.A., Acar, B.S., Bez, H.E.: Two Dimensional Statistical Linear Discriminant Analysis for Real-Time Robust Vehicle Type Recognition. In: SPIE, vol. 6496 (2007)
19. Zhao, W., Chellappa, R., Phillips, P.J., Rosenfeld, A.: Face recognition: A literature survey. ACM Computing Surveys 35(4), 399–458 (2003)