Vehicle subtype, make and model classification from side profile video

Conference or Workshop Item

Accepted Version

Vehicle Subtype, Make and Model Classification from Side Profile Video

Jonathan Boyle and James Ferryman
Computational Vision Group, School of Systems Engineering, University of Reading, UK
{j.n.boyle | j.m.ferryman}@reading.ac.uk

Abstract

This paper addresses the challenging domain of vehicle classification from pole-mounted roadway cameras, specifically from side-profile views. A new public vehicle dataset is made available consisting of over 10000 side profile images (86 make/model and 9 sub-type classes). 5 state-of-the-art classifiers are applied to the dataset, with the best achieving high classification rates of 98.7% for sub-type and 99.7-99.9% for make and model recognition, confirming the assertion made that single vehicle side profile images can be used for robust classification.

1. Introduction

Vehicle classification has many important applications including traffic analysis, tolling, and law enforcement such as border checkpoints where it is of interest to verify the identity of vehicles which may have spoofed number plates. This work focusses on classification from side profile views of vehicles. The paper first describes the approach to detect the passage of a vehicle and extract a suitable normalised image ready for classification. This process results in a new public dataset of over 10000 images made available to the wider research community. Second, a number of state-of-the-art classifiers are evaluated to validate the use of side profile images for vehicle sub-type, make and model recognition.

2. Related work

Prior work on video-based vehicle classification is generally limited to splitting observed vehicles into a small number of categories. The approaches of [10, 13] estimate the size of detected vehicles, with [10] exploiting camera calibration to classify vehicles as truck/non-truck. Instead of considering just size, [6] use more statistics of the foreground silhouettes to classify seven classes (cars, vans and various sizes of trucks), but notably not the different subtypes of cars, nor their makes and models. One approach to extending beyond simple size estimates is to consider fitting simple wireframe 3D models to the observed vehicles [3]. The advantage of such model based approaches is a reduction in view-point dependence; however, they remain limited to the basic classes (bus/lorry, van, car/taxi, motorbike/bicycle) and producing simple 3D models accurate enough to distinguish between the many makes and models or subtypes of private cars seems unlikely to have high success. Turning to exploiting lower level image features, [7] apply a two-step kNN classifier with geometric and texture based features for 7 types classes of vehicle exploiting both frontal and rear views, [5] applied various classifiers to three vehicle type classes based on SURF and Gabor features, and [12] developed an SVM classifier based on a structural edge signature extracted from rear vehicle views. However, [12] is limited to 3 classes with a total of 1664 images. Investigation of make and model classification is generally a more recent occurrence. [15] apply number plate detection to captured frontal images of vehicles and train a classifier ensemble based on the Gabor transform and PHoG for 600 images/15 brands/21 vehicle classes. A similar viewpoint is used by [8], which considers edge, gradient and corner features and report 96% make/model accuracy based on a small set of 262 frontal vehicle images. [4] and [14] provide further review of various approaches.

This work departs from these approaches to consider side profile views and a significantly larger number of classes. For vehicle subtypes, the side profile is considered to be the most discriminating viewpoint, however it is also conjectured that it provides a large quantity of information that can be used for make and model classification in terms of the shapes of side windows, presence and location of trim, as well as overall vehicle shape. The vehicle wheels also provide a more reliably detectable feature for normalisation than is possible using the licence plate and frontal views. Finally, for the installation location, it was found that with flowing and queueing traffic, the frontal view of vehicles

This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no. 312784. We would also like to thank Murray Evans for his preliminary input to this work.

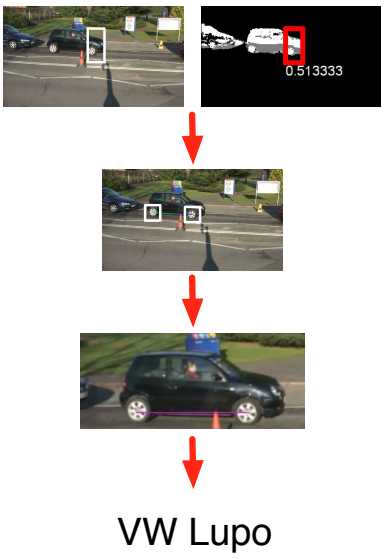

Figure 1: Detection and classification process. The vehicle enters the scene, and background subtraction occurs. Foreground pixels in the "trigger zone" trigger vehicle detection. Wheels are detected, then used for normalisation. The normalised image is classified.

was often occluded, which the side view did not suffer from.

3. Side Profile Dataset

A camera was installed to monitor the Pepper Lane entrance of the University of Reading with optical axis perpendicular to the flow of traffic at a height of approximately 2.5m. At this height, the vehicle is seen primarily in profile, and the visibility of roof, bonnet (hood), and boot (trunk) surfaces is minimised. The camera was set to record two hours of the morning rush-hour every working day over the course of four weeks. Once vehicles had been detected and extracted, the resulting dataset contained over 10000 images of vehicles, most of which were manually classified into sub-types as well as over 86 make/model categories with a wide range of populations. This data, as well as the frames extracted when vehicles are present, and the full normalised and labelled training set, are made available at `http://www.cvg.reading.ac.uk/rvd`.

3.1. Vehicle detection

Detection of the vehicle's presence in the acquired images is performed using standard background subtraction [16]. A region towards the middle of the image (see Figure 1) is specified as the "trigger zone", and when the ratio of foreground to background pixels within this region exceeds 0.25 the image is considered to contain a vehicle. At this stage, wheel detection is performed. An overview of the detection and normalisation process is shown in Figure 1.

3.1.1 Wheel Detection

Wheel detection is performed using a trained wheel/not-wheel classifier and a stochastic, multi-dimensional search. First, a Random Forest classifier [2] was trained on 27674 wheel images (and 13643 negative (non-wheel) random images) manually extracted from the dataset, totalling 41319 images. Each wheel training image is a square image centred on the centre of the wheel, with a width to approximately match the diameter of the wheel plus some of the surround (the presence of wheel-arch and some body-work is considered beneficial context to discriminate a wheel from the background or other clutter). The forest is grown in a traditional manner with three decision nodes (*RGB colour difference*, *Gradient magnitude*, *Gradient direction*, see Section 4.1.1) that analyse the low-level features. One of the nodes is selected at random, and then its parameters (pixel locations, thresholds, etc.) optimised to produce the best decision as measured by the Gini-index. The final forest contained 798 trees. Because the image features are extremely simple, the forest can classify a large number of image locations in a reasonable period of time allowing the wheel search to be conducted in near real-time.

The wheel search seeks to the optimal location in the image for two wheels however, not just one. As such, there are five dimensions, x, y, d_x, d_y, s, where (x, y) describes the location of the rear wheel, $(x + d_x, y + d_y)$ the location of the front wheel, and s controls the size of the bounding box extracted from the image and used by the wheel classifier (although only a single lane of traffic is observed, the variation in wheel size between vehicle types (e.g. SUV, city car) can be large enough to warrant the search in scale.). The search is initialised to an estimated location relative to the "trigger zone" which is based on initial observations, or to the previously detected location if the trigger remains active. Letting v_w represent the votes of the classifier for the wheel class, and $v_{\bar{w}}$ represent the votes for non-wheel class, the search seeks to maximise $v_w - v_{\bar{w}}$ for both wheels.

An example of a wheel detection can be seen in the central image of Figure 1, while Figure 2 shows some examples resulting from the normalisation process for both the RGB colour image and the foreground mask image.

3.1.2 Normalization

Classification is a far simpler task if the input data is first normalised to account for scale and rotation variations. Assuming that the most forward and most rearward wheels of the vehicle can be detected in the image, simple transformations are applied to scale and rotate the detected vehicle to produce a normalised image. Once the wheels have been detected, a 200×85 pixel image is created that places the front wheel at $(150, 70)$ and the rear wheel at $(50, 70)$ by translation, rotation, scaling and cropping of the original

Figure 2: Examples of the normalised RGB and foreground mask composing the vehicle dataset.

sub-type	image count
city	882
hatchback	5821
large hatchback	694
saloon	1074
estate	680
people carrier	215
sports	189
SUV	357
van	589

Table 1: Vehicle sub-types and the total number of labelled images for each

image. As the background subtraction process provides a useful shape descriptor in the form of foreground mask, this too is normalised and made available for the classification process.

The database of manually annotated images available for classification consists, in total, of more than 10000 images divided in a highly unbalanced way into 9 vehicle sub-types and 86 make/model classes (see Table 3) as typically understood in the UK, as shown in Tables 1 and 4.

4. Vehicle Classification

Classification is performed using five available classifiers:

- Img-Forest: A random forest operating on simple image features.
- Evo-Forest: An evolutionary forest that fuses the random forest with with techniques from genetic algorithms to evolve a forest optimised for the classification task.
- HoG-Forest: A random forest operating on a HoG descriptor of the whole image. Decision nodes are the more traditional selecting one attribute of the descriptor and comparing to a threshold.
- HoG-Lin-SVM: A linear SVM operating on the image's HoG descriptor.
- HoG-RBF-SVM: An SVM with an RBF kernel operating on the image's HoG descriptor.

4.1. Classifiers

4.1.1 Random Forest

The Random Forest, introduced by Breiman[2], is an ensemble of binary decision trees where each decision node of the tree is configured to exploit a randomly chosen classifier from a set of available classifiers, with randomly initialised parameters optimised to produce the best split of the training examples. It represents an appealing classifier because of its natural multi-class nature, and accompanying fast computation. In this work the vehicle dataset consists of RGB images and black/white foreground masks. The colour images are simply processed to produce gradient magnitude and gradient orientation images (these are not thresholded to produce edge images). This enables a number of possible decision nodes that can be considered:

1. *RGB colour difference*: Given two pixels of the RGB image, threshold the Euclidean colour difference. This requires 5 parameters as the image coordinates of the two pixels (x_1, y_1), (x_2, y_2) and τ, the threshold. (CIELAB colour space was also tested for this node, but there was no appreciable difference in classifier performance).
2. *Gradient magnitude*: Given a single pixel coordinate (x, y), determine if the image gradient has a magnitude larger than a threshold τ.
3. *Gradient orientation*: Given a single pixel coordinate (x, y), determine if the orientation of the image gradient is between two thresholds τ_1 and τ_2
4. *Mask pixels*: Given a single pixel coordinate (x, y), determine if the pixel is foreground or background in the mask image.
5. *Scale range*: When normalising the image, the scale factor applied to resize the image can be recorded. Assuming all vehicles are a similar distance from the camera, this will provide an estimate of the vehicle size. The node checks if the scale is between two thresholds.

The generated forests had a total of 200 grown trees each, with a maximum depth of 20. When a node is first being created, a type from the list above is selected at random and optimised using the Gini Impurity.

4.1.2 Evolutionary Forest

While each tree of the Random Forest is optimised to be the best classifier it can be, the trees are not optimized to work together to produce the best forest. Hence this work also considers an evolutionary approach to growing the forest applying a fitness function that aims to ensure each new tree maximises its own strength, while minimising correlation with existing trees. The decision nodes are the same as for the random forest. The process of training an evolutionary forest consisting of n trees summarised as follows:

1. Start with an empty set F_f of trees.
2. Iterate until $|F_f| = n$:
   (a) Create a sub-sample of the training set for training trees of this pool.
   (b) Grow a pool of potential trees F_p.
   (c) Iterate until ready:
      i. evaluate trees in pool
      ii. replace poor trees with new trees by cross-breeding/mutating/growing
   (d) take best tree in F_p and add to F_f

A maximum number of 200 trees is imposed, with a terminating condition to stop adding additional trees if the accuracy over the training set reaches 100%. Further details of the evolutionary forest implementation are given in [1].

4.1.3 HoG-based Random Forest

A further random forest was considered whereby each image is represented by a pre-processed Histogram of Oriented Gradients (HoG) vector. With the vehicle images normalized to images of size (200x85) a HoG descriptor of 7524 elements is generated with a block size of (20,14) pixels and a block stride of (10,7) pixels from a greyscale version of the image. Instead of using multiple nodes as described in Section 4.1.1, only a single type of node was used which compares a single value in the vector to a threshold.

4.1.4 HoG-based SVM classifiers

The Support Vector Machine (SVM) is a favoured classifier in the literature. For this work, the one-vs-all multi-class approach is used (as defended by [9]) where a classifier is trained for each class against all other classes. The winning class for a given unknown input is the classifier with the largest positive result. The same HoG vector as described in Section 4.1.3 is used to train both an SVM with a linear kernel and an RBF kernel. For each final SVM trained, a cross-validation grid search is performed for each SVM to determine the optimal parameters. The linear SVM's C parameter was optimised in the range $\{10, 100, 1000\}$. The RBF SVM, with the same C parameter range, had a search on γ in the range $\{0.0001, 0.001, 0.01\}$.

5. Experiments

5.1. Metrics

Each classifier was trained and tested 20 times. Multiple statistics (average accuracy, recall, precision) for multi-class classification, as described in [11], have been calculated to measure the performance, with the minimum, median and maximum for each statistic recorded (see Tables 3 and 2):

$$Average\ Accuracy = \frac{\sum_{i=1}^{c} \frac{tp_i + tn_i}{tp_i + fn_i + fp_i + tn_i}}{c} \quad (1)$$

$$Recall_M = \frac{\sum_{i=1}^{c} \frac{tp_i}{tp_i + fn_i}}{c} \quad (2)$$

$$Precision_M = \frac{\sum_{i=1}^{c} \frac{tp_i}{tp_i + fp_i}}{c} \quad (3)$$

5.2. Vehicle Subtype

The vehicle dataset was first divided into classes as shown in Table 1 based on expert judgement, which amounts to a common interpretation of the types of cars on UK roads. The dataset was divided into eight cross-validation training/testing sets, where labelled images were selected at random to be either training or testing. The training sets were set to be at most $\min(0.5n, 200)$ images (where n is the total number of available training images for the class), with the remainder being used for testing.

5.3. Vehicle Make/Model

To see what effect there might be on varying the number of classes the make/model dataset was split into 4 subsets based on the number of vehicles available per class. A summary of these splits are shown in Table 4. Again, the available data was split such that 50% of the images, or at most 200 images, were used for training, the remainder for testing. This did mean that the training data for these sets could be quite un-balanced, with some sets providing 200 training images and some a mere 10.

where i is the class number, c is the number of classes, tp_i, tn_i, fp_i and fn_i are respectively the number of true positives, true negatives, false positives, and false negatives, and M indicates the macro-average across all classes [11].

		Set 20			Set 40			Set 100			Set 300		
		Min	Med	Max	Min	Med	Max	Min	Med	Max	Min	Med	Max
Img-Forest	Acc	0.993	0.994	0.994	0.992	0.992	0.993	0.986	0.987	0.988	0.960	0.965	0.971
	Rec	0.570	0.602	0.623	0.675	0.698	0.713	0.766	0.782	0.797	0.879	0.897	0.912
	Prec	0.737	0.782	0.817	0.794	0.812	0.829	0.825	0.845	0.861	0.872	0.891	0.912
Evo-Forest	Acc	0.997	0.998	0.998	0.995	0.996	0.997	0.989	0.990	0.992	0.963	0.971	0.976
	Rec	0.895	0.907	0.922	0.883	0.895	0.904	0.900	0.911	0.920	0.911	0.927	0.940
	Prec	0.843	0.858	0.883	0.829	0.849	0.883	0.866	0.887	0.908	0.903	0.924	0.938
HoG-Forest	Acc	0.996	0.996	0.997	0.994	0.995	0.995	0.994	0.994	0.995	0.985	0.986	0.989
	Rec	0.843	0.851	0.857	0.875	0.880	0.887	0.949	0.953	0.956	0.977	0.979	0.983
	Prec	0.830	0.843	0.847	0.833	0.850	0.860	0.823	0.854	0.860	0.613	0.616	0.625
HoG-Lin-SVM	Acc	0.999	0.999	**0.999**	0.999	0.999	**0.999**	0.998	0.998	**0.998**	0.991	0.995	**0.997**
	Rec	0.933	0.947	**0.953**	0.936	0.949	**0.961**	0.966	0.975	**0.981**	0.973	0.983	**0.990**
	Prec	0.943	0.953	**0.961**	0.942	0.950	**0.956**	0.967	0.970	**0.975**	0.972	0.982	**0.989**
HoG-RBF-SVM	Acc	0.999	0.999	**0.999**	0.999	0.999	**0.999**	0.998	0.998	**0.998**	0.966	0.969	0.976
	Rec	0.929	0.937	0.948	0.939	0.945	0.953	0.967	0.974	0.980	0.881	0.899	0.921
	Prec	0.937	0.949	0.956	0.942	0.948	**0.956**	0.959	0.969	0.973	0.911	0.924	0.940

Table 2: Comparison of classification results for five classifiers on the Make/Model data splits (see Table 3). Highest numerical values for each statistic are highlighted in bold.

		Min	Med	Max
Img-Forest	Acc	0.944	0.948	0.954
	Rec	0.843	0.857	0.874
	Prec	0.661	0.697	0.727
Evo-Forest	Acc	0.965	0.973	0.979
	Rec	0.826	0.838	0.855
	Prec	0.697	0.743	0.774
HoG-Forest	Acc	0.970	0.975	0.979
	Rec	0.882	0.888	0.900
	Prec	0.776	0.809	0.819
HoG-Lin-SVM	Acc	0.978	0.982	0.985
	Rec	0.950	0.957	**0.963**
	Prec	0.837	0.855	0.886
HoG-RBF-SVM	Acc	0.979	0.982	**0.987**
	Rec	0.950	0.958	**0.963**
	Prec	0.837	0.866	**0.889**

Table 3: Comparison of classification results for five classifiers on the Subtype data splits (see Table 1). Highest numerical values for each statistic are highlighted in bold.

# available	# classes	# images
300	6	3032
100	27	6360
40	59	8305
20	86	9141

Table 4: Make/Model data splits, showing the number of classes and total number of images used for training for classes that have at least the specified number of images available.

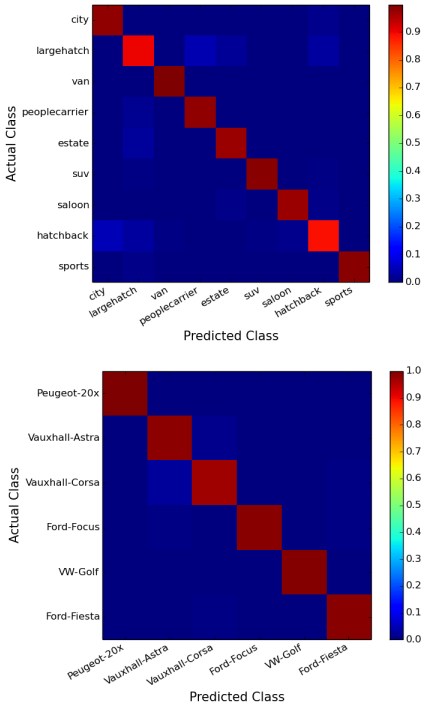

Figure 3: Confusion Matrices. Top: Subclass HoG-RBF-SVM, Bottom: Make/Model Set 300 HoG-Lin-SVM. Best viewed in colour.

5.4. Results

Analysing the results for both subtype and make and model (Tables 3 & 2, Figure 3), it is clear that the two SVM classifiers provide effectively equivalent results and superior classification performance to the three forest classifiers, with the RBF kernel outperforming the linear kernel for subtype and vice-versa for make and model. It is also clear that using the HoG descriptor provides a more robust classification than using a set of low-level image features (Img-Forest). This can be explained by the HoG descriptor's blocks providing some robustness against small misalignments in the normalisation process and a lower susceptibility to corruption caused by other sources of image noise. It is also noteworthy that the make and model classification obtains higher accuracy than that of the sub-type classification task, however this mostly can be explained by intra-class and inter-class variation. Most vehicle makes and models exhibit quite small intra-class variation, though

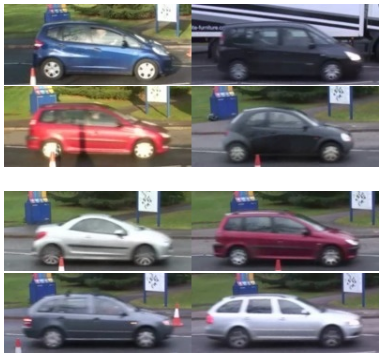

Figure 4: Misclassifications. Left column: Instance, right column: Classification. Best viewed in colour.

there are some exceptions such as long running marques like the Ford Fiesta that have undergone several generations and 'face lifts'. This means that the classifier can generally learn very specific details for discriminating between the different classes, and even though some make/model classes may be quite similar (see Figure 4, rows 3 & 4), the stability of each class' appearance permits the classifier to find the distinctive features to tell them apart. This is far more difficult with the subtypes classification, where not only is the intra-class variation quite large (as each class covers a wide range of vehicle makes and models), but the inter-class variation can be quite small. Indeed, the exact point where one category ends and another begins can often be quite subjective. There will always be vehicle models that span multiple categories. These can be subtle issues such as there being a number of vehicle models that retain a saloon shape but which actually have a hatchback style rear opening (classified as saloons in this work), or more troublesome like the issue of large 5-seat family cars designed to be very much the same shape and style as full size 7-seat people carriers (1st row, Figure 4), or where exactly the distinction lies between a small city car and a more general purpose hatchback (2nd row, Figure 4). While the expert annotator employed best judgement, there remains subjectivity and overlap between the classes.

6. Conclusions and Future Work

This work has presented a method for detecting and classifying vehicles into sub-type and make and model from side profile views and produced a public database of labelled data. Furthermore, high classification rates of 98%-99% are obtained from single images extending the state-of-the-art. Further work will establish if the obtained high classification rates are retained as the number of make/model classes increases and the inter-class variation between vehicles decreases. Additionally, colour classification will be considered as well as performance of the detection/classification system when it is run live.

References

[1] Vehicle classification using evolutionary forests. In *Proc. of International Conference on Pattern Recognition Applications and Methods*, pages 387–393, 2012.

[2] L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

[3] N. Buch, J. Orwell, and S. Velastin. Urban road user detection and classification using 3d wire frame models. *Computer Vision, IET*, 4(2):105–116, 2010.

[4] N. Buch, S. A. Velastin, and J. Orwell. A review of computer vision techniques for the analysis of urban traffic. In *IEEE Trans. on Intelligent Transportation Systems*, volume 12, pages 920–939, September 2011.

[5] P. Dalka and A. Czyżewski. Vehicle classification based on soft computing algorithms. In *Lecture Notes in Computer Science*, volume 6086, pages 70–79, 2010.

[6] C. Huang and W. Liao. A vision-based vehicle identification system. In *Pattern Recognition, 2004. ICPR 2004. Proc. of the 17th Int. Conf. on*, volume 4, pages 364–367, 2004.

[7] N. C. Mithun, N. U. Rashid, and S. M. M. Rahman. Detection and classification of vehicles from video using mulltiple time-spatial images. In *IEEE Trans. Intelligent Transportation Systems*, volume 13, pages 1215–1225, September 2012.

[8] G. Pearce and N. Pears. Automatic make and model recognition from frontal images of cars. In *Advanced Video and Signal Based Surveillance (AVSS), 2011 8th IEEE Int. Conf. on*, pages 373–378, 2011.

[9] R. Rifkin and A. Klautau. In defense of one-vs-all classification. In *Journal of Machine Learning Research*, volume 5, pages 101–141, 2004.

[10] S. Shi, Z. Qin, and J. Xu. Robust algorithm of vehicle classification. In *Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2007. SNPD 2007. Eighth ACIS International Conference on*, pages 269–272, 2007.

[11] M. Sokolova and G. Lapalme. A systematic analysis of performance measures for classification tasks. *Inf. Process. Manage.*, 45(4):427–437, July 2009.

[12] N. S. Thakoor and B. Bhanu. Structural signatures for passenger vehicle classification in video. In *IEEE Trans. on Intelligent Transportation Systems*, volume 14, pages 1796–1805, 2013.

[13] R. Wang, L. Zhang, K. Xiao, R. Sun, and L. Cui. Easisee: Real-time vehicle classification and counting via low-cost collaborative sensing. In *IEEE Trans. on Intelligent Transportation Systems*, volume 15, pages 1–11, 2014.

[14] K. Yousaf, A. Iftikhar, and A. Javed. Comparative analysis of automatic vehicle classification techniques: A survey. *Int. Journal of Image, Graphics and Signal Processing*, 9:52–59, 2012.

[15] B. Zhang. Reliable classification of vehicle types based on cascade classifier ensembles. *Intelligent Transportation Systems, IEEE Trans. on*, 14(1):322–332, 2013.

[16] Z. Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *Pattern Recognition, 2004. 17th Int. Conf. on*, pages 28–31, 2004.