



Provided by the author(s) and University of Galway in accordance with publisher policies. Please cite the published version when available.

Title	Multi-Class and Single-Class Classification Approaches to Vehicle Model Recognition from Images
Author(s)	Madden, Michael G.; Munroe, Daniel T.
Publication Date	2005
Publication Information	"Multi-Class and Single-Class Classification Approaches to Vehicle Model Recognition from Images", Daniel Munroe and Michael G. Madden. Proceedings of AICS-05: Irish Conference on Artificial Intelligence and Cognitive Science, Portstewart, Sept 2005.
Item record	http://hdl.handle.net/10379/191

Downloaded 2023-09-29T04:05:28Z

Some rights reserved. For more information, please see the item record link above.



Multi-Class and Single-Class Classification Approaches to Vehicle Model Recognition from Images

Daniel T. Munroe and Michael G. Madden

Department of Information Technology
National University of Ireland, Galway
Galway, Ireland

{daniel.munroe, michael.madden}@nuigalway.ie

Abstract. This paper investigates the use of machine learning classification techniques applied to the task of recognising the make and model of vehicles. Although a number of vehicle classification systems already exist, most of them seek only to distinguish between vehicle categories, e.g. identifying whether a vehicle is a bus, truck or car. The system presented here demonstrates that a set of features extracted from the frontal view of a vehicle may be used to determine the vehicle type (make and model) with high accuracy. The performance of some standard multi-class classification algorithms is compared for this problem. A one-class k-Nearest Neighbour classification algorithm is also implemented and tested.

1 Introduction

The need for vehicle identification and classification technologies has become relevant in recent years as a result of increased security awareness for access control systems in parking lots, buildings and restricted areas. Vehicle recognition can also play an important role in the fields of road traffic monitoring and management. For example, in the automatic toll collecting systems on roads, vehicles have to be classified into categories in order to calculate the correct amount to charge.

Vehicle type recognition, as a process of identifying the correct make and model from a frontal image of a vehicle (car), represents a natural extension of conventional number-plate recognition systems. Number-plate recognition software could benefit from the system proposed in this paper, by providing a double-check to combat the problem of fake number plates.

The recognition process proposed in this paper is based on using specific feature extraction techniques from digital images. Different machine learning algorithms are tested on the dataset of 150 frontal view images of vehicles (30 images of each of five classes), and experiments are carried out to assess their performance.

Two broad approaches to machine learning classification are considered: multi-class classifiers and single-class classifiers. As discussed below in Section 3, multi-class classification is the ‘standard’ approach used in machine learning, but the single-class approach is more appropriate in some applications where standard assumptions about the distribution of examples do not apply.

After providing a brief overview of related research and the concept of single-class classification, the system is described in more detail. Then, the performance of various classification algorithms is analysed, and conclusions are drawn.

2 Related Research

Various approaches to vehicle classification and detection have been reported in the computer vision literature. Despite the large amount of literature in vehicle detection, there has been relatively little done in the field of vehicle classification. It is a relatively challenging problem due to the wide variety of vehicle shapes and sizes, making it difficult to categorise vehicles using simple parameters.

Most systems either detect (locate a vehicle against a background) or classify vehicles into broad categories such as cars, buses and trucks [3, 5, 7, 8, 14, 16]. Wei *et al.* [14] use a 3-D parameterised model which corresponds to features of the vehicle's topological structure, classified using a neural network. They present results showing that 91% of the vehicles are correctly identified into six different categories. Lipton *et al.* [8] describe a vehicle tracking and classification system that can classify moving objects as vehicles or human beings, but its purpose is not to separate vehicles into different classes. Their system obtained over 86% correct classification on vehicles and 83% correct on humans.

Gupte *et al.* [5] present an algorithm for detection and classification of vehicles in image sequences of traffic scenes. The system classifies vehicles into two categories – cars and non-cars (e.g. buses, trucks, SUV's). In a 20-minute sequence of highway traffic, 90% of the vehicles were correctly detected and tracked, and of these correctly tracked vehicles, 70% were correctly classified. Kato *et al.* [7] propose the development of a driver assistance system using a vision-based preceding (vehicles travelling in the same direction as the subject vehicle) vehicle recognition method, which is capable of recognising a wide selection of vehicle types against road environment backgrounds. The classification method they used is the multiclustered modified quadratic discriminant function. The system classifies vehicles into three different categories and has a success rate of 97.7%.

Dubuisson Jolly *et al.* [3] use a deformable template algorithm consisting of finding a template that best characterises the vehicle into one of five categories. Their algorithm was tested on 405 image sequences and had a recognition rate of 91.9%. Similarly, Yoshida *et al.* [16] describe a local-feature based vehicle classification system, which classifies vehicles using a computer graphics model. They use a template matching technique and achieve a 54% accuracy rate, when classifying the images into five categories.

More strongly related work to ours, in terms of what is being achieved, is that of Petrović *et al.* [11] demonstrate that a relatively simple set of features extracted from frontal car images can be used to obtain high performance verification and recognition of vehicle types. Recognition is initiated through an algorithm that locates a region of interest (ROI) and using direct or statistical mapping feature extraction methods, obtains a feature vector, which is classified using a nearest neighbour algorithm.

They state that the system is capable of recognition rates of over 93% when tested on over 1000 images containing 77 different classes.

3 Single-Class Classification

All of the systems described in the previous section are based on multi-class classifiers. Multi-class (including two-class) classification is the standard approach used in machine learning, whereby a hypothesis is constructed that discriminates between a fixed set of classes. For example, a classifier may distinguish between images that either show a vehicle or do not, or distinguish between trucks, buses, vans and cars. However, multi-class approaches make two assumptions:

1. Closed set: all possible cases fall onto one of the classes
2. Good distribution: the training set is composed of cases that are statistically representative of each of the classes

While these assumptions do not appear onerous, they may or may not be reasonable in practice. For example, the closed-set assumption is valid when classifying images as having a vehicle present or not present in them, but may not be valid when classifying vehicles into categories (what about tractors, motorbikes and heavy machinery?) Conversely, when classifying vehicles into categories, the distribution assumption may be valid as it is straight-forward to acquire images that are representative of each category, but it might not be valid for the task of distinguishing vehicles from non-vehicles—should the counter-example images show just empty roads, or people, animals, birds, buildings, bicycles, trees and other subjects?

As machine learning researchers and practitioners in recent years have tackled problems where these assumptions are not valid, because for some classes there is either no data, insufficient data or ill-distributed data available, techniques for single-class classification have begun to receive some attention. Essentially, such techniques form a characteristic description of the target class, using this to discriminate it from any other classes (which are considered outlier classes). Clearly, this avoids the closed-set assumption, and also does not require the availability in the training data of statistically representative samples of classes other than the target class.

The first algorithms for single-class classification were based on neural networks, such as those of Moya *et al.* [9, 10] and Japowicz *et al.* [6]. More recently, one-class versions of the support vector machine have been proposed, notably by Tax [13] and Scholkopf *et al.* [12]. Tax's approach is to find the smallest volume hypersphere (in feature space) that encloses most of the training data. Scholkopf *et al.* aim to find a binary function that takes the value +1 in a "small" region capturing most of the data, and -1 elsewhere. They transform the data so that the origin represents outliers, and then find the maximum margin separating hyperplane between the data and the origin. Scholkopf *et al.* note that both methods are equivalent in some circumstances.

In this paper, we use a simple single-class classification technique based on the k-Nearest Neighbour (kNN) algorithm. The single-class kNN algorithm was chosen because, as will be discussed in Section 4.1, in our initial experiments comparing multi-

class classification algorithms it was found that the multi-class kNN worked well. In this algorithm, a test object is classified as belonging to the target class when its local density is larger or equal to the local density of its nearest neighbour in the training set (target class) [13].

The single-class kNN classifier has a number of parameters that may be adjusted; the number of neighbours can be changed so that the average k distances to the first k neighbours is calculated; the threshold value of accepting outlier classes may be changed; also, the distance metric may be changed. Figure 1 shows an example of a target class consisting of Ford Focuses. The algorithm for detecting whether or not a test case A (e.g. Volkswagen Golf) is in the target class is shown immediately below.

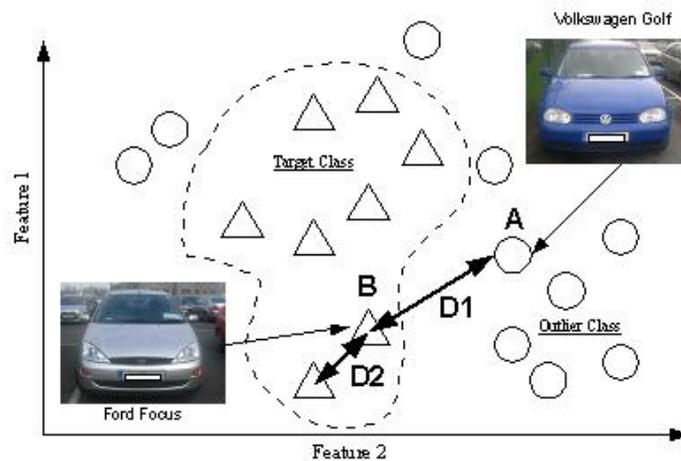


Fig. 1. One-class k -nearest neighbour classifier applied to vehicle recognition dataset

One-class k -nearest neighbour classification algorithm

To classify A as a member/not member of target class

1. Set a threshold value (e.g. 1) and choose the number of k distances
2. Find nearest neighbour for A in the target class, call this B and call the distance D1
3. If $k = 1$
 - Find the nearest neighbour for B in the target class and call this distance D2
 - Else
 - Find the average distances to the k -nearest neighbours for B in the target class and call this distance D2
4. If $D1 / D2 > \text{threshold value}$
 - Reject A as a target class
 - Else
 - Accept A as a target class

3 Vehicle Type Recognition

For this work, a dataset of frontal images of vehicles was compiled over a period of several weeks, and reflect a range of weather and lighting conditions. The dataset is made up of 150 images of vehicles — 30 images of each of five classes. The classes are: Opel Corsa, Ford Fiesta, Ford Focus, Volkswagen Polo and Volkswagen Golf. Naturally, care was taken to include only one version of each vehicle make/model, as for example the 1998 Golf would have to be considered as a different class from the 2004 Golf, since these two versions have quite different appearances. All images contain frontal views of a single vehicle captured from slightly different distances and from a height of approximately 1 metre. The images have 1600 x 1200 colour pixels. A sample of each class of car is shown in Figure 2.



Fig. 2. Examples of the five different car types as they appear in the dataset

The system is implemented in Matlab using the Image Processing Toolbox. The image is converted to a grayscale image and automatically cropped to exclude the top half. The next step is to detect edges in the image. Edge detection highlights sharp changes in intensity, as differences in intensity can correspond to the boundaries of the features in the image. After experimenting with some alternative algorithms, the Canny edge detection [2] method was chosen because it succeeded in finding all the important features in the image. The Canny edge detector first smoothes the image using a Gaussian filter to eliminate noise before performing the edge detection. Dilation was then used to fill the gaps left by the edge detector. Dilation is an operation that “grows” or “thickens” objects in a binary image and is controlled by a shape referred to as a linear structuring element [4].

After having reduced the image to a series of edges, standard elements of the image such as the lights and license plate are identified automatically. A fixed-length numerical feature vector is then derived for each vehicle, representing geometric properties of the various elements of the image.

Finally, as described in next, different machine learning classifiers are used to determine the vehicle make and model associated with each vector. The overall procedure is illustrated in Figure 3.

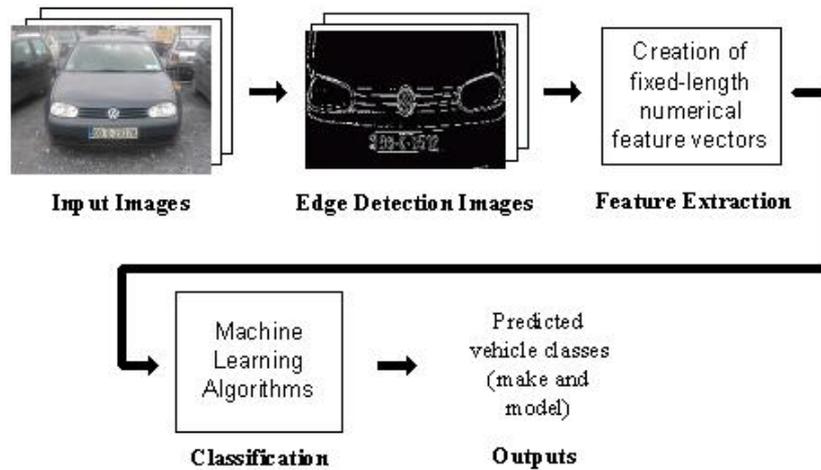


Fig. 3. Overall System for Vehicle Make/Model Identification

4 Experimental Results

Two sets of experiments have been performed. The first set of experiments is described below in Section 4.1. They involved comparing the performance of a range of standard multi-class classifiers on the dataset, since multi-class classifiers have been used in previous approaches to vehicle identification/classification. Previous approaches have used different forms of feature extraction, so it is interesting to consider how our approach to feature selection works with standard classifiers. The specific classification algorithms chosen are the C4.5 decision tree, the k -nearest neighbour classifier and a feed-forward neural network trained using backpropagation. The implementations of these in the WEKA machine learning package [15] were used. The default settings in WEKA for these algorithms were used.

The purpose of the second set of experiments is to evaluate the performance of a single-class classifier for this task. The single-class kNN algorithm that has been described in Section 3 was implemented in Matlab and its performance evaluated as discussed in Section 4.2.

4.1 Multi-Class Classification Results

Figure 4 compares the learning curves of the three multi-class classification algorithms under consideration. A learning curve gives an indication of the amount of data required to achieve good performance with a classification algorithm. It is constructed by randomly sampling training sets from the overall dataset, at a range of percentages between 5% and 90% of the overall dataset. Each time, a classifier is constructed with

the training data set and evaluated on the remainder of the data. This procedure is repeated 10 times for each training set size and the results averaged.

The learning curves indicate that classification performances of the k -nearest neighbour and neural network algorithms are comparable with each other, and better than that of the decision tree algorithm, at least at lower training set sizes. The curves also show that 70% of the dataset is sufficient for 100% classification accuracy using kNN or the neural network.

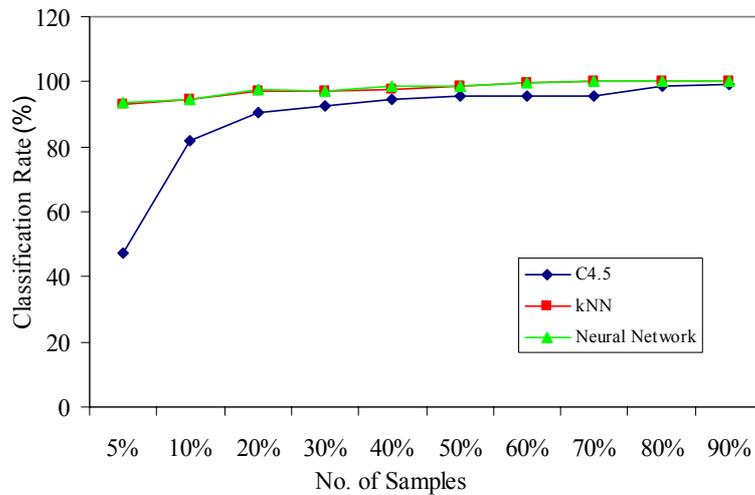


Fig. 4. Comparison of Learning Curves for the Multi-Class Classification Algorithms

The performance of each algorithm was also evaluated using 10 x 10-fold sorted cross-validation [1]. Using this technique, the data is divided randomly into ten parts, each part is held out in turn and the learning scheme trained on the remaining nine-tenths. The procedure is repeated ten times and the average for the ten parts is calculated. The whole process is repeated for ten different runs and the average and standard deviation is calculated. Table 1 lists the accuracy (average \pm standard deviation) on the training data of each of the three multi-class classification algorithms, computed using a 10 x 10-fold cross-validation.

Although the results for kNN are numerically higher than those of the other two algorithms, a paired t-test based on the 10 x 10-fold sorted cross-validation runs did not identify the difference as being statistically significant at the 5% significance level.

Table 1. Results of the 10 x 10-fold cross-validation

Algorithm	Accuracy (%)
C4.5	98.53 ± 3.34
KNN	99.99 ± 0.21
Neural Net	99.53 ± 1.47

4.2 One-class Classification Results

The target class contains 20 examples of numerical feature vectors representing a certain vehicle (e.g. Opel Corsa) and the test set contains 134 examples of numerical feature vectors of different images of vehicles (10 of the target class, 30 of each of the other 4 class types and 4 unknowns). The 4 unknowns are images of cars not in the dataset (e.g. Toyota Corolla). For each target class the process is repeated a number of times and the average is calculated. The results obtained are shown in Table 2. The performance of the kNN one-class classifier algorithm also depends on a number of predefined choices, as stated above. The k value can be changed so that the average k distances to the first k neighbours are calculated. The best-fitting value of k calculated is $k = 1$. The threshold value of accepting outlier classes can also be changed. The best threshold evaluation carried out from experiments is 1.5. The one-class classifier predicted a high percentage of correctly identifying target and outlier classes, but a downfall to the method is calculating the threshold value for optimising high performance levels.

Table 2. Results of the one-class nearest neighbour classifier

Target Class	Composition of Test Set	Test Set Accuracy
Opel Corsa (20 examples)	134 examples (10 Opel Corsa, 30 of each of the other 4 classes and 4 unknowns)	98.50%
Ford Fiesta (20 examples)	134 examples (10 Ford Fiesta, 30 of each of the other 4 classes and 4 unknowns)	95.02%
Ford Focus (20 examples)	134 examples (10 Ford Focus, 30 of each of the other 4 classes and 4 unknowns)	98.50%
VW Golf (20 examples)	134 examples (10 VW Golf, 30 of each of the other 4 classes and 4 unknowns)	97.26%
VW Polo (20 examples)	134 examples (10 VW Polo, 30 of each of the other 4 classes and 4 unknowns)	98%
Average:		97.46%

5. Conclusions

Vehicle recognition is an important technology for developing systems for road traffic monitoring and management and security issues. However, it is difficult task for computer systems to achieve because vehicles have a wide range of different appearances due to the variety of their shapes and colours.

This paper proposes a novel vehicle recognition process that identifies the vehicle make and model (e.g. Volkswagen Golf) from a frontal image. Extracted fixed-length numerical feature vectors are tested and classified using different machine learning techniques. Of the multi-class classifiers considered, the kNN and the neural network classifiers appear to be most effective for this task, with accuracy of over 99.5%.

A single-class kNN classifier was also evaluated, as single-class classifiers have the benefit of not making assumptions about having a closed set of classes or having a training data set that is fully representative of data that would be encountered in practice. This classifier was also shown to perform well, with an overall accuracy rate of about 97.5%.

Clearly, it is not reasonable to draw direct comparisons between the results of the multi-class and single-class classifiers presented here, as the experimental methodology and assumptions underlying are quite different. In particular, we note that multi-class results could be made arbitrarily bad by adding vehicle types to test set that do not appear in the training set (since the multi-class classifier output cannot represent the concept ‘none of the above’), whereas this should not be detrimental to the performance of the single-class classifier. Other approaches could be used to defend against this problem, for example using two-class classifiers and training them using a one-versus-all classification scheme. However, such an approach would not be theoretically well-motivated, as the negative examples would represent a diverse collection of classes, and would still not be statistically representative of the negative concept.

In the future, we propose to assess the performance of other forms of single-class classifier on this problem domain. We also intend to accumulate a library of vehicle images that do not fall into any of the classes considered here.

References

1. Bouckaert, R.R.: Choosing between two learning algorithms based on calibrated tests. International Conference on Machine Learning (ICML-2003), Washington DC, 2003
2. Canny, J.: A Computational Approach to Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-8, (6): 679-698, 1986
3. Dubuisson Jolly, MP., Lakshmanan, S., Jain, A.K.: Vehicle Segmentation and Classification using Deformable Templates. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 18, No. 3, March 1996
4. Gonzalez, R.C., Woods, R.E., Eddins, S.L.: Digital Image Processing using Matlab. Prentice Hall, 2004

5. Gupte, S., Masoud, O., Martin, R.F.K. Papanikolopoulos, N.P.: Detection and Classification of Vehicles. IEEE Transactions on Intelligent Transportation Systems, Vol. 3, No. 1, March 2002
6. Japkowicz, N., Myers, C., Gluck, M.: A novelty detection approach to classification. Proc. 14th International Joint Conference on Artificial Intelligence, Montreal, Canada, 1995
7. Kato, T., Ninomiya, Y., Masaki, I.: Preceding vehicle recognition based on learning from sample images. IEEE Transactions on Intelligent Transportation Systems, Vol. 3, No. 4, December 2002
8. Lipton, A.J., Fujiyoshi, H., Patil, R.S.: Moving target classification and tracking from real-time video. IEEE Workshop Applications of Computer Vision, 1998, pp. 8-14
9. Moya, M., Koch, M., Hostetler, L.: One-class classifier networks for target recognition applications. In Proceedings World Congress on Neural Networks, Portland, OR, International Neural Network Society, INNS, pp. 797-801, 1993
10. Moya, M.R. & Hush, D.R., 1996: "Network Constraints and Multi-Objective Optimisation for One-Class Classification." Neural Networks, Vol. 9, No. 3.
11. Petrović, V., Cootes, T.: Analysis of features for rigid structure vehicle type recognition. BMVC2004, 2004
12. Scholkopf, B., Platt, J.C., Shawe-Taylor, J., Smola, A.J., Williamson, R.C.: Estimating the support of a high-dimensional distribution. Neural Computation, Vol. 13. (Also a Microsoft Research technical report, MSR-TR-99-87, 1999.) 2001
13. Tax, D.M.J.: One-class classification. PhD Thesis, Delft University of Technology. ISBN: 90-75691-05-x, 2001
14. Wei, W., Zhang, Q., Wang, M.: A method of vehicle classification using models and neural networks. IEEE Vehicular Technology Conference, IEEE, 2001
15. Weka Homepage: The Waikato Environment for Knowledge Analysis. Department of Computer Science, University of Waikato, NZ, <http://www.cs.waikato.ac.nz/ml>
16. Yoshida, T., Mohottala, S., Kagesawa, M., Ikeuchi, K.: Vehicle Classification Systems with Local-Feature Based Algorithm using CG Model Images. IEICE Trans., Vol. E00-A, No. 12, December 2002