Barry, T., & Nagarajah, C. R. (2010). Object recognition in industrial environments using support vector machines and artificial neural networks.

Originally published in:
International Journal of Advanced Manufacturing Technology, 48(5–8), 815–821.
Available from: http://dx.doi.org/10.1007/s00170-009-2313-3

Object Recognition in Industrial Environments using Support Vector Machines and Artificial Neural Networks

T. J. Barry C. R. Nagarajah

Abstract

This paper presents a comparison between Artificial Neural Networks and Support Vector Machines in the application of classifying automotive wheels in an industrial environment. Performance of these two approaches over a range of classifier parameters on a dataset pre-processed in multiple ways has been evaluated and the results analysed. Results indicate that the best performance is obtained using a Support Vector Machine approach incorporating a linear kernel.

Keywords: Artificial Neural Networks, Support Vector Machines, Feature Extraction and Wheel Identification.

1 Introduction

Automatic classification by visual inspection is a useful tool in a manufacturing environment. The use of a fixed camera for inspection is of particular interest in cases where the natural movement of the part to be inspected is highly repeatable and passes through an area with sufficient room to mount lighting and cameras such that they are free from obstruction. The benefits of automated inspection include high repeatability, consistent accuracy, and ability to detect faults that are not visible to a human operator.

The specific application of wheel identification is particularly suited to camera based inspection due to: the possibility of achieving low image noise and strong feature edges with back lighting; repeatable wheel position; perceivable rotational variability in the wheel limited to the degree of rotational symmetry; a limited and unique set of wheels to select between. The challenge in this application, as discussed by Shabestari [1], comes from choosing the correct feature set for each wheel so that even the most similar of wheel types is correctly classified every time and developing a robust enough classifier to cope with noise and typical dimensional variations observed in manufactured products.

Support Vector Machines (SVM) and Artificial Neural Networks (ANN) have been used in many applications including: bearing fault detection [2]; breast cancer cell detection [3]; drug classification [4]; image retrieval [5]; identifying students with learning disabilities [6]; modelling a microwave transmitter [7]; protein fold recognition [8]; signature recognition [9]; and textile colour classification [10].

These investigations indicate that the SVM algorithm will generally perform better than an ANN, however there are exceptions which occur when the user has specific knowledge of the application and the available data [4]. When SVMs are outperformed by expertly trained ANNs, the performance difference between the classifiers is usually small [2, 11].

The problems related to using vision based systems in a manufacturing environment are significant. The main challenge has been identified as designing a robust system that can function within a noisy environment and accurately determine a result repeatedly within strict timing requirements. For a vision system, this usually means being capable of dealing with system drift (accumulated small changes), vibration, lighting changes, and other random noise. Also, in terms of achieving a long lasting solution, the process of retraining to include new products should be a relatively simple task that does not compromise performance or lead to any unnecessarily large amounts of down time.

2 Support Vector Machines

Support Vector Machines were first introduced by Vapnik [12, 13], however the tutorials presented by Burges [14] and Law [15] provide a very informative introduction to this topic, as are the more detailed descriptions presented by Cristianini & Shawe-Taylor [16, 17] and Gunn [18].

SVMs seek to determine a linear separator between binary data classes. The optimal position of the separating plane is specified as where the margin between the plane and the data points is maximised. This concept is generally referred to as a Maximal Margin Classifier, and its strength lies in determining a good general solution to the classification problem without overfitting the data.

In order to separate more complex datasets, a kernel function is used to map the data into a higher dimensioned space where a single hyperplane can separate the binary classes. The commonly used kernels tested in this work included:

- Linear;
$$K(\mathbf{x}, \mathbf{y}) = \mathbf{x}.\mathbf{y} \tag{1}$$
- Polynomial;
$$K(\mathbf{x}, \mathbf{y}) = (a\mathbf{x}.\mathbf{y} + b)^d \tag{2}$$
- Radial Basis Functions; and
$$K(\mathbf{x}, \mathbf{y}) = e^{-\gamma\|\mathbf{x}-\mathbf{y}\|^2}, \quad \gamma > 0 \tag{3}$$
- Sigmoid which is the basis for the standard Multi-Layered Perceptron.
$$K(\mathbf{x}, \mathbf{y}) = \tanh(\kappa\mathbf{x}.\mathbf{y} - \delta) \tag{4}$$

Multi-class classification can be achieved using either a multi-class version of the SVM algorithm, or by the more common method of using a combination of binary classifiers with a decision function. Crammer & Singer [19] and Weston & Watkins [20] present multi-class SVMs, and compare their performance against combinations of binary SVMs resulting in performance similar to the algorithms listed below, with some additional computational load. In this paper, the notation mSVM will refer only to the multi-class classifiers achieved using combinations of binary SVMs. Common mSVM choices include [21–23]:

- One versus One (OvO)
- One versus All (OvA)
- Direct Acrylic Graph (DAG)
- Error Correcting Output Codes (ECOC)
- Binary Tree Architecture (BTA)

Research [24, 25] appears to indicate that generally one algorithm will perform slightly better than the others on a given application, however the degree of improvement is usually very small. As Anthony [26] states, the main contender for best general mSVM performance is the OvO algorithm. And so in this work, the OvO approach has been used as it has been shown to perform well in every test. Also, as a small percentage of the data is used for training at a given time, OvO is generally quite fast to train and classify even though it usually requires significantly more classifier evaluations than OvA. As classes are not combined in the training stage, pre-classification and class grouping are not required for the OvO approach.

The support vector machine algorithm has been implemented using the LIBSVM toolbox in MATLAB[1] [27]. LIBSVM is written in the C++ language and accessed via a MATLAB EXecutable (MEX) function. The optimal kernel and parameters values have been determined by testing parameters in the range of $2^{-15}, 2^{-14}, \ldots, 2^{5}$, except for the polynomial order which was tested in the range of $2, \ldots, 8$.

Parameter estimation and training have been completed using the Leave-One-Out (LOO) approach to reduce the potential problems inherent in having few samples for training compared to the amount of data in each sample.

3 Artificial Neural Networks

ANNs are parallel systems modelled on the biological processes in the human brain. There are numerous examples of ANNs in pattern recognition and data classification [28]. ANNs in this work have been implemented using the MATLAB toolbox. The parameters to consider in the design stage are the network layout, the number of hidden layers, the number of neurons in each of the hidden layers, and the network training method. Based on the algorithm benchmarks presented in [29] a feed-forward network with backpropagation and one hidden layer was chosen. The following training methods were selected:

- Levenberg-Marquardt Backpropagation Algorithm (ANN-LM)
- Resilient Backpropagation Algorithm (ANN-RB)
- Polak-Ribiére Conjugate Gradient Algorithm (ANN-PRCG)
- Scaled Conjugate Gradient Algorithm (ANN-SCG)

These methods have been chosen as they generally performed the best on pattern matching application as opposed to the function approximation application.

To avoid the problem of overfitting, the number of neurons in the hidden layer ($\#h$) was given an upper limit of the number of ANN inputs [29] and a lower limit of the number of classes.

4 Data Preprocessing

The first problem to be addressed in image processing is to extract the features of interest from the source image. To reduce the noise effects, the symmetrical nature of the automotive wheel has been used to identify those features that occur regularly around the image. This averaging approach, however does remove some of the sharper edges as the backlighting varies across the image, leaving dark patches.

The image processing and feature extraction approach used was: firstly to improve the image quality by balancing the lighting with histogram equalisation; secondly, convert the image to binary with a maximum entropy adaptive threshold; and finally perform open and close morphologies to remove bit noise. Next the centre hole of the wheel was detected using morphological comparisons, in this case, a white circle with a 34 pixel radius compared to a black band with inner radius of 35 pixels and outer radius of 45 pixels. This combination was found to only exist simultaneously at the centre hole, so the image could be re-centred and cropped to a square.

Before removing the centre hole and the mounting bolt holes in the wheel, which were common to all images and hence unnecessary for classification, the degree of symmetry was estimated using known possible angle rotations about the centre which consist of the set $\alpha(S) = \frac{360^\circ}{S}$ where $S \in \{5, 6, \ldots, 9\}$.

[1] http://www.mathworks.com/

Table 1: Feature dataset contents

Dataset	Small Image Fig. 1d	Full Image Fig. 1a	Symmetry From Fig. 1c	Number of Features From Fig. 1c	Radial Histogram Fig. 1e	Ellipse Axes From Fig. 1c
ImageData1	×					
ImageData2		×				
ImageData3	×		×	×	×	
FeatureData1			×	×	×	
FeatureData2			×	×	×	×
FeatureData3			×	×		×
Feature Size	900	307200	1	1	240	36

To detect asymmetric noise, each image was rotated through each symmetric angle $\alpha(S)$. Any image pixel that occurred more than half the time was designated as signal, and the rest noise. The noise was detected before removing the centre hole so that there was a sharp edge in the radial histogram (in polar coordinates from the image centre) between the centre hole and the wheel image.

The number of independent features around the wheel were counted and bounding ellipses were fitted to each. For each ellipse the major and minor axes were estimated [30]. And finally, one section of symmetry was extracted from the image, the image downsampled keeping one pixel in every 8×8 pixel region, and lastly the edge is detected by subtracting the image from a one pixel eroded version of itself.

The final data processing stage involved normalising values across each feature data set to reduce the occurrence of poor classifier training due to poorly scaled data. For example, each histogram was scaled to within the range $[0, 1]$. And the set of the number of features was scaled so that the largest number of features (eighteen) was recorded in the feature dataset as 1.

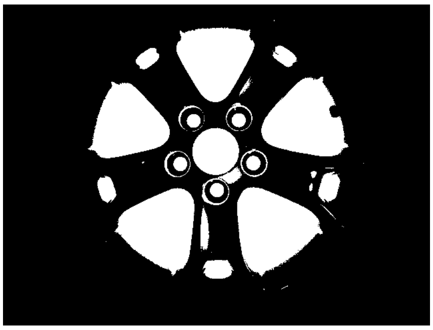
The net result of all of this processing is an image reduced from (640×480) to (30×30) pixels, a radial histogram, a set of minor and major elliptical axes, the degree of symmetry and the number of features. However, as the importance of each of these features is unknown before processing, several versions of the datasets were created, and are summarised in Table 1.

To demonstrate the image processing results on one of the images in the set, Fig. 1 includes the main sequential steps from raw image to final output. Figs. 1a-c and 2a-i have been shrunk for display purposes by a factor of four.

The image dataset consists of 41 images belonging to 9 different wheel types, a sample of each has been included in Fig. 2. The number of classifier classes was chosen so that each class represents a single wheel type. The number of images in each class is not evenly distributed due to scarcity of data, for example, class 1 has two images, while class 3 has ten images and so using the LOO training and validation approach certain classes were trained on a single dataset. To compensate for this deficiency in the training data, a weight matrix has been constructed for each class to give training priority to those samples from the less well populated classes. The weights were chosen such that the most populated class has weight 1, and the other classes have weights calculated by:

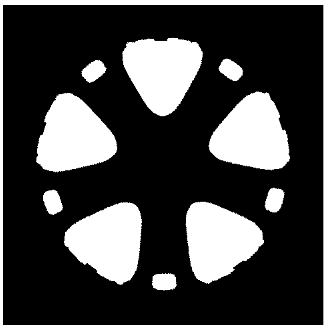
$$\text{weight}(i) = \left(\frac{\text{maximum number of images in any class}}{\text{number of images in class } i} \right)$$

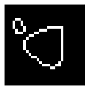
The images were taken from two different wheel identification (ID) cells, each with a slightly different camera offset and position with respect to the wheel resulting is a slight scaling difference between data collected from the two cells. The perceivable difference between the images from each wheel ID cell is negligible after image processing.



a) Original image

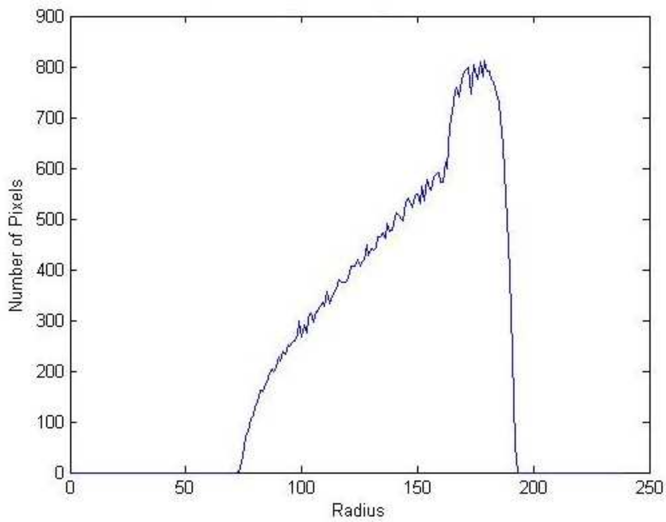
b) Image after histogram equalisation and max entropy thesholding





c) Image after symmetrical feature detection and noise removal

d) Final reduced image with only edges remaining



e) Radial histogram of Fig. 1c before normalising

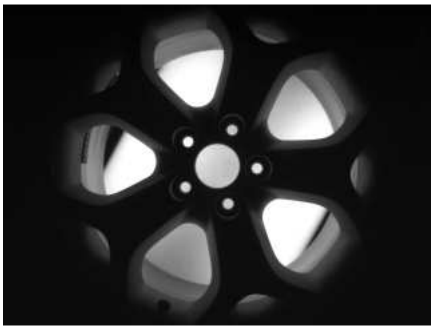
Figure 1: Image treatment cycle for wheel type 1

a) Wheel type 1

b) Wheel type 2

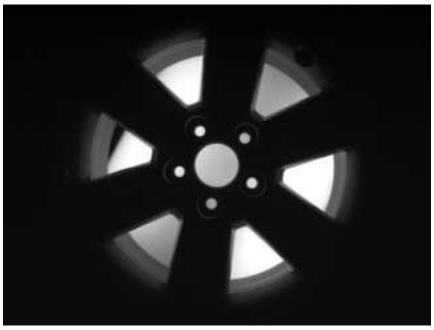
c) Wheel type 3

d) Wheel type 4



e) Wheel type 5

f) Wheel type 6



g) Wheel type 7

h) Wheel type 8

i) Wheel type 9

Figure 2: Sample images for each wheel type

Table 2: Best results for each classifier type

Classifier	**Best Dataset**	**Parameters**	**Training Set Accuracy**	**Validation Accuracy**	**Classification Time** (s)
ANN-LM	FeatureData3	$\#h = 19$	96.3%	92.7%	0.296
ANN-RB	FeatureData3	$\#h = 30$	93.5%	90.2%	0.231
ANN-PRCG	FeatureData1	$\#h = 51$	92.2%	92.7%	0.232
ANN-SCG	FeatureData3	$\#h = 22$	92.6%	91.8%	0.363
SVM-Lin	FeatureData3	$C = 2^{-10}$	100%	100%	0.221
SVM-Poly	FeatureData3	$C = 2^{-5},\ a = 2^{-10},\ b = 2^{-15},\ d = 2$	100%	100%	0.103
SVM-RBF	FeatureData3	$C = 2,\ \gamma = 2^{-12}$	100%	100%	0.097
SVM-Sig	ImageData1	$C = 4,\ \kappa = 2^{-5},\ \delta = 2^{-15}$	100%	90.2%	0.501

5 Results

This work has generated a substantial amount of information regarding the performance of the ANN and SVM algorithms on the datasets under a wide range of parameters and with different training methodologies. For example, when choosing parameters for a RBF mSVM, two parameters are required, C, which is the SVM misclassification tolerance parameter, and γ as defined in Eq. 3. The LOO classification results are presented in Fig. 3. A logarithmic axis has been chosen for the parameters as the tested values increased in powers of 2.

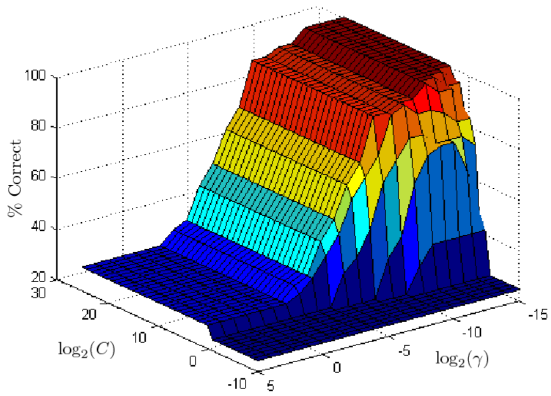


Figure 3: LOO classification accuracy for changing SVM parameters

The best validation results for each classification methods are as shown in Table 2. The set of parameters specified in column three of Table 2 are, for all ANN approaches, the number of neurons in the hidden layer ($\#h$), and for the SVM approaches are defined in Eqs. 1, 2, 3, and 4 except for C which is the SVM misclassification tolerance parameter. When the best result consisted of multiple parameter values, the classifier with the shortest computation time was chosen. The computational equipment was a dual core 2.2GHz Pentium processor with 2GB RAM running the Windows XP SP2 operating system and MATLAB release 2008b.

As shown in Table 2, the performance of the mSVM, excluding SVM-Sig, in classifying automotive wheel types is consistently better than that of the ANNs in terms of speed and classifier accuracy. These results do not take into account the computational requirements of the image processing stage as that is a common load for all classifiers. Furthermore the processing time taken to identify the best set of parameters for each of the algorithms has been omitted.

The most interesting results are for SVM-Sig which was the only classifier to achieve its best result with image data rather than feature data, and as the image processing stage attached to extracting the ellipse properties is the most computationally expensive part of the image processing stage, then this preference for ImageData1 actually results in a computational saving of approximately twenty seconds per image compared to all classifiers requiring FeatureData1&3.

However, little work has been done to optimise the computational loading of the image processor, so it is conceivable that this advantage is less than it appears. In several of the SVM solutions, equal performance was measured with both FeatureData2 and FeatureData3, indicating an ability to extract the useful features from the more information rich datasets.

6 Conclusion

Investigation as described here indicate that Support Vector Machines outperform Artificial Neural Networks, both in terms of classification accuracy and classification speed. Except in one case, SVMs and ANNs trained on features extracted from the data were found to provide higher classifier accuracy that those trained on the images themselves.

Future work will include expanding the number of classes to classify; testing performance when in the presence of a variety of artificially introduced noise patterns; incorporating principal component analysis into the data preprocessing stage; extracting a wider variety of features from each image to highlight other image and statistical properties; and exploration of the performance of the Artificial Neural Network approach with a wider range of training algorithms and more hidden layers.

Acknowledgements

The authors would like to thank the Ford Motor Company of Australia and the Australian Research Council for their support for this project through the linkage grant scheme.

References

[1] B. N. Shabestari, J. W. V. Miller, and V. Wedding. Wheels identification using machine vision technology. In *IEEE International Conference on Systems Engineering*, pages 273–275, 1991.

[2] B. Samanta, K. R. Al-Balushi, and S. A. Al-Araimi. Artificial neural networks and support vector machines with genetic algorithm for bearing fault detection. *Engineering Applications of Artificial Intelligence*, 16:657–665, 2003.

[3] H. X. Liu, R. S. Zhang, F. Luan, X. J. Yao, M. C. Liu, Z. D. Hu, and B. T. Fan. Diagnosing breast cancer based on support vector machines. *Journal of Chemical Information and Computer Sciences*, 43:900–907, 2003.

[4] E. Byvatov, U. Fechner, J. Sadowski, and G. Schneider. Comparison of support vector machines and artifical neural network systems for drug/nondrug classification. *Journal of Chemical Information and Computer Sciences*, 43:1882–1889, 2003.

[5] W. T. Wong and S. H. Hsu. Application of SVM and ANN for image retrieval. *European Journal of Operational Research*, 173:938–950, 2006.

[6] T. K. Wu, S. C. Huang, and Y. R. Meng. Evaluation of ANN and SVM classifiers as predictors to the diagnosis of students with learning disabilities. *Expert Systems with Applications*, 34:1864–1856, 2008.

[7] N. Türker and F. Güneş. A competitive approach to neural device modeling: Support vector machines. *Lecture Notes in Computer Science*, 4132:974–981, 2006.

[8] C. H. Q. Ding and I. Dubchak. Multi-class protein fold recognition using support vector machines and neural networks. *Bioinformatics*, 17(4):349–358, Apr 2001.

[9] E. Frias-Martinez, A. Sanchez, and J. Velez. Support vector machines versus multi-layer perceptrons for efficient off-line signature recognition. *Engineering Applications of Artificial Intelligence*, 19:693–704, 2006.

[10] R. Furferi and L. Governi. The recycling of wool clothes: an artificial neural network colour classification tool. *The International Journal of Advanced Manufacturing Technology*, 37(7–8):722–731, 2008.

[11] F. Liang. An efficient bayesian neural network classifier with a comparision study to support vector machines. *Neural Computation*, 15:1959–1989, 2003.

[12] V. N. Vapnik. *Statistical Learning Theory.* Adaptive and Learning Systems for Signal Processing, Communications, and Control. John Wiley & Sons, 1^{st} edition, 1998.

[13] V. N. Vapnik. *The Nature of Statistical Learning Theory.* Statistics for Engineering and Information Scientists. Springer, 2^{nd} edition, 1999.

[14] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.

[15] M. Law. A simple introduction to support vector machines. Lecture for CSE 802, Dept. of Comp. Sci. and Eng., Michigan State University, 2006.

[16] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and other kernel-based learning methods.* Cambridge University Press, 2000.

[17] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis.* Cambridge University Press, 2004.

[18] S. R. Gunn. Support vector machines for classification and regression. Technical report, Faculty of Engineering, Science and Mathematics, School of Electronics and Computer Science, University of Southhampton, May 1998.

[19] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:265–292, 2001.

[20] J. Weston and C. Watkins. Multi-class support vector machines. Technical Report CSD-TR-98-04, Department of Computer Science, Royal Holloway University of London, May 1998.

[21] K. B. Duan and S. S. Keerthi. *Which is the Best Multiclass SVM Method? An Empirical Study.*, volume 3541/2005 of *Lecture Notes in Computer Science*, chapter Performance Analysis, pages 278–285. Springer, 2005.

[22] B. Fei and J. Liu. Binary tree of SVM: A new fast multiclass training and classification algorithm. *IEEE Transactions on Neural Networks*, 17(3):696–704, May 2006.

[23] C. W. Hsu and C. J. Lin. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425, Mar 2002.

[24] S. Cheong, S. H. Oh, and S. Y. Lee. Support vector machines with binary tree architecture for multi-class classification. *Neural Information Processing - Letters and Reviews*, 2(3):47–51, Mar 2004.

[25] D. Tsujinishi, Y. Koshiba, and S. Abe. Why pairwise is better than one-against-all or all-at-once. *Proceedings of the IEEE International Joint Conference on Neural Networks*, 1:693–698, Jul 2004.

[26] G. Anthony, H. Gregg, and M. Tshilidzi. Image classification using SVMs: One-against-One vs One-against-All. In *Proccedings of the 28th Asian Conference on Remote Sensing*, 2007.

[27] C. C. Chang and C. J. Lin. LIBSVM: a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/∼cjlin/libsvm.

[28] C. M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, 1995.

[29] H. Demuth, M. Beale, and M. Hagan. *Neural Network Toolbox, User's Guide.* The MathWorks, Inc., 2007. Version 5.

[30] R. Halif and J. Flusser. Numerically stable direct least squares fitting of ellipses. Technical report, Department of Software Engineering, Charles University, Czech Republic, 2000.