Automatic Traffic Scene Analysis Using Supervised Machine Learning Algorithms - Backpropagation Neural Networks and Support Vector Machines

Heejong Suh[1], Daehyon Kim[2*] and Changsoo Jang[3]

[1] Department of Electronic Communication Engineering, Chonnam National University, Dundeok-Dong, Yeosu-Si, Jeollanam-Do, Korea
[2] Department of Marine and Civil Engineering, Chonnam National University, Dundeok-Dong, Yeosu-Si, Jeollanam-Do, Korea
[3] Department of Computer Engineering, Chonnam National University, Dundeok-Dong, Yeosu-Si, Jeollanam-Do, Korea
**Corresponding author. E-mail address: daehyon@chonnam.ac.kr*

Abstract

Automatic traffic scene analysis which has been used for real-time on-road vehicle detection system is essential to many areas of ITS (Intelligent Transport Systems). In order to improve the detection time and accuracy of detection performance, various image processing techniques have been used for real-time vehicle detection. Moreover, Neural Networks have been increasingly and successfully applied to many problems for ITS research topics. Support Vector Machines (SVMs) are currently another efficient approach to vehicle detection because of their remarkable performance. In this research, two different models, Backpropagation which is the best-known neural network model and SVMs have been studied to compare their performance in predictive accuracy, through experiment with real world image data of traffic scenes. Experimental results show that SVMs can provide higher performance in terms of predictive performance than the well-known Backpropagation neural network model.

Keywords: *ITS (Intelligent Transport Systems), Neural Networks, SVM (Support Vector Machines), vehicle detection, Backpropagation*

1. Introduction

Computer vision that extracts traffic information from the road image collected by a video camera is one of the main Intelligent Transportation Systems (ITS) research subjects. It is a key element for integrated automatic traffic surveillance and control systems on roads. Currently, various types of detection techniques have been developed for automatic traffic data collection - vehicle counting, vehicle speed detection, vehicle tracking, congestion and incident detection. However, image-processing techniques for collecting traffic data are potentially more powerful and flexible than the methods currently available, such as the buried induction loop detector, microwave, or infra-red beams.

Since the late 1980's, there has been a growing interest in the application of computer vision system in transportation engineering. However, algorithms for collecting traffic data are still not sufficiently robust and reliable in the complicated circumstances, and many studies have been done to develop more efficient and reliable algorithms. Until recently, Artificial Neural Networks were regarded as an efficient algorithm for automatic traffic data collection using road video images. Even though various types of neural network models have been developed, the multilayer feed-forward using the Backpropagation learning algorithm has been one of the most popular neural networks in transportation engineering research areas since it has been applied successfully to various problems, and yielded a relatively good performance.

However, after SVMs (Support Vector Machines) were introduced by Vapnik [1], they have recently received a great deal of attention, because of their remarkable performance. SVMs are generation learning systems based on advances in statistical learning theory, and have been successfully applied to numerous pattern recognition problems, including object detection [2], handwritten character recognition [3, 4, 5], text categorization [6], face detection in images [7], and traffic scene analysis [8]. In this paper, two learning models of Backpropagation and SVMs are used for vehicle detection in an image processing-based traffic surveillance system. The vehicle detection performance of two learning models is compared in edge-detected images of real world traffic scenes, and this study then proposes the best model for real world traffic scene analysis.

2. Backpropagation and SVMs (Support Vector Machines)

3.1. Backpropagation Neural Networks

The multilayer feed-forward learning algorithm using Backpropagation [9] is one of the most popular neural network models. Even though it is widely and successfully applied to various problems, there are some problems to be solved in the standard Backpropagation model such as the expensive computing cost for training, lack of rule for the proper selection of the network topology, and possibility of being trapped at a local minimum during the training process. In order to solve the problems of the standard Backpropagation, many extensions and modifications have been considered in previous researches [10, 11].

In this paper, the BMP (Backpropagation with Momentum and Prime offset) model, which is an advanced Backpropagation model, has been used, since it has been shown that the BMP model is more efficient than other Backpropagation models in terms of prediction accuracy and computing cost for training [11]. The BMP model is the combination of Backpropagation with momentum [9] and Quickprop [10]. In Quickprop, Fahlman suggested the Prime-offset parameter be the derivative of a sigmoid function, in order to solve the flat-spots problem. The Prime-offset parameter can be applied to the standard Backpropagation or the Backpropagation with Momentum model. The BMP (Backpropagation with Momentum and Prime-offset) model adds the Prime-offset parameter to the Backpropagation with Momentum model [11].The process of the BMP model can be summarized as follows:

Step 1: Initialize synaptic weights

Step 2: Present the input and output vectors

Step 3: Calculate output values for each unit in the network by using

$$Out_{pk}^{l} = f_{pk}^{l}(Net_{pk}^{l}) \quad \text{where} \quad Net_{pk}^{l} = \sum_{j=1}^{K_{l-1}} w_{kj}^{l} In_{pj}^{l} + \theta_{k}^{l} \tag{1}$$

In equation (1), In^{l}_{pj} are inputs to the k^{th} neuron in the layer *l*, *w* is the total number of synaptic weights in the network, θ^{l}_{k} is a bias term, K_l is the number of *l* layer nodes, *p* is a training pattern and $f()$ is a transfer function.

Step 4: For the output layer, $l = L$, calculate the values of weight changes using

$$\Delta_p w^{L}_{kj} = \eta \delta^{L}_{pk} In^{L}_{pj} \quad \text{where} \quad \delta^{L}_{pk} = \left(y_{pk} - Out^{L}_{pk}\right)\left(f^{L}_{k}(Net^{L}_{pk})\right)' + \text{Prime offset} \tag{2}$$

In equation (2), y_{pk} is the desired output.

Step 5: For the hidden layers, $l = 1, ..., L$-1, calculate the values of weight changes using

$$\Delta_p w^{l}_{ju} = \eta \delta^{l}_{pj} In^{l}_{ij} \ , \quad \text{where} \quad \delta^{l}_{pj} = \left(f^{l}_{j}(Net^{l}_{pj})\right)' \sum_{k=1}^{K_{l+1}} \delta^{l+1}_{pk} w^{l+1}_{kj} \tag{3}$$

Step 6: Update weights on the output layers, by

$$w^{L}_{kj}(t+1) = w^{L}_{kj}(t) + \Delta_p w^{L}_{kj} \tag{4}$$

Step 7: Update hidden layers by

$$w^{l}_{ju}(t+1) = w^{l}_{ju}(t) + \Delta_p w^{l}_{ju} \quad \text{for } l = 1,..., L\text{-}1 \tag{5}$$

Step 8: Repeat these steps, until the average squared error computed over the entire training set is at an acceptably small value. The error for the output units is then calculated by

$$Err_p = \sum_{k=1}^{K_L} (y_{pk} - Out^{L}_{pk})^2 \tag{6}$$

3.2. Support Vector Machines

Support vector machines introduced by Vapnik [1] are based on statistical learning theory. SVMs are primarily designed for two-class classification problems, and have been expanded to a multi-class classifier. The basic idea of the algorithm for the two-class classification problem is as follows [12]:

Let the training data $(\mathbf{x}_i, y_i)$, for $i = 1,\dots,l$, be $y_i \in \{\pm 1\}$ $\mathbf{x}_i \in \mathbf{R}^N$. Then the support vector algorithm simply looks for the optimal hyper-plane with the largest margin. This can be formulated as follows:

$$\min \ \tau(\mathbf{w}, \xi) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l} \zeta_i \tag{7}$$

$$\text{s. t. } \ y_i(\mathbf{x}_i \cdot \mathbf{w} + b) \geq 1 - \xi_i \ , \quad i = 1,\dots,l \tag{8}$$

where **w** is a normal to the hyper-plane, *b* is the bias or offset, and *C* is the upper bound for the Lagrange multiplier, λ_i, i.e., $0 \leq \lambda_i \leq C$. The primal form of the objective function can be:

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{l} \alpha_i (y_i((\mathbf{x}_i \cdot \mathbf{w}) + b) - 1) \tag{9}$$

The Lagrangian *L* has to be minimized with respect to the primal variables **w** and *b* and maximized with respect to the dual variables α_i. From the Karush-Kunhn-Tucker conditions, the derivative of L with respect to the primal variables must vanish (Fletcher, 1987), subject to the constraints $\alpha_i \geq 0$, i.e.,

$$\frac{\partial}{\partial \mathbf{w}} L(\mathbf{w}, b, \boldsymbol{\alpha}) = 0, \ \frac{\partial}{\partial b} L(\mathbf{w}, b, \boldsymbol{\alpha}) = 0 \tag{10}$$

Equation (4) leads to

$$\mathbf{w} = \sum_{i=1}^{l} \alpha_i y_i \mathbf{x}_i \ \text{ and } \sum_{i=1}^{l} \alpha_i y_i = 0 \tag{11}$$

Equation (11) is equality constraints in the dual formulation, and the following equation (12) which is the Wolfe dual of the optimization problem is given by substituting them into equation (9).

$$\max \ W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j) \tag{12}$$

$$\text{s.t. } \alpha_i \geq 0, \text{ and } \sum_{i=1}^{l} \alpha_i y_i = 0 \tag{13}$$

The hyperplane decision function can thus be given as

$$f(x) = \operatorname{sgn}\left((\mathbf{x} \cdot \mathbf{w}) + b\right) = \operatorname{sgn}\left(\sum_{i=1}^{l} y_i \alpha_i (\mathbf{x} \cdot \mathbf{x}_i) + b\right) \tag{14}$$

The algorithm explained above is for linearly inseparable classification problem. For nonlinear classification problems, SVMs are using kernels. The most popular kernel is the Gaussian kernel of equation (15)

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \left\| \mathbf{x}_i - \mathbf{x}_j \right\|^2) \tag{15}$$

where γ determines the width of the kernel function.

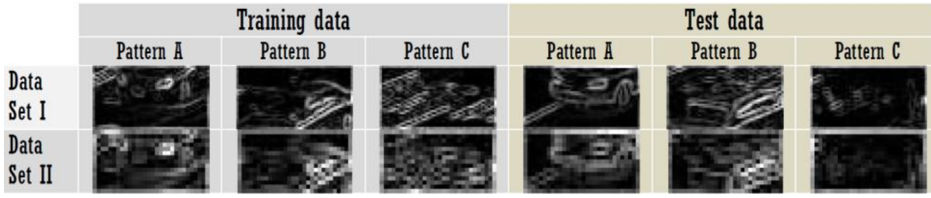
3. Experiments and results

3.1. Data Sets for Learning and Testing

For the experiments in this study, Sobel edge detected images with three different patterns have been used: Pattern A, Pattern B, and Pattern C (see Figure 1). The Pattern A images correspond to the top of a vehicle, while Pattern B images are the rear part of a vehicle and Pattern C are non-vehicle images on the roads. From the traffic scene, a total of 930 data sets have been obtained, 400 sets for Pattern A, 400 sets for Pattern B and 130 sets for Pattern C. The total data sets were split into two subsets, one for training and the other for testing (see Table 1).

Table 1. Data sets for training and test

Patterns	Number of training data	Number of test data	Total data
Pattern A	100	300	400
Pattern B	100	300	400
Pattern C	30	100	130
Total data	230	700	930



Note: Data Set I is Sobel edge-detected images of 30 pixel by 60 pixel size. Data Set II is the shrunken images of Data Set I obtained by selecting every other pixel, and are 15 pixel by 30 pixel sized images.

Figure 1. Three patterns and experimental images for training and test

3.2. Network Architecture and Parameter Value

For application of the Backpropagation model, this study used a three layers network, *i.e.*, one-hidden-layer network, since the previous papers [13, 14, 15, 16] showed that only one-hidden-layer network was adequate to produce a good performance for the pattern recognition problem. In this paper, two types of network topology have been used in order to cope with different number of input units, *i.e.*, 450(input units) – 225(hidden units) – 3(output units) and 120(input units) – 60(hidden units) – 3(output units).

The implementation of Backpropagation was for batch mode learning with a learning rate of 0.1, a momentum of 0.95, and a prime-offset of 0.1. For the output vectors, three units were used to recognize three different patterns; i.e. $y_{pattern1}$ = [1 0 0], $y_{pattern2}$ = [0 1 0], and $y_{pattern3}$ = [0 0 1]. The sigmoid function was used for the activation function, and all networks were fully interconnected, i.e. input layer to hidden layer, and hidden layer to output layer. The stopping criterion of the network training was that 100% recognition accuracy be achieved on the training set.

For the application of the SVMs model, two parameters, C and γ, should be determined in advance. The parameter C is a positive regularization parameter that controls the tradeoff between the complexity of the machine and the allowed classification error, and γ is the parameter of the Gaussian kernel of equation (15). For the parameter *C*, all experiments have been carried out in this study with $C = 1.0$, since it has been shown that the value of this parameter does not affect the predictive performance [8]. However, the parameter γ of the Gaussian RBF kernel may affect the predictive performance, and the experimental results show a different performance according to the parameter value. In this study, the best prediction performance was achieved with Gamma (γ) = 0.02 and Gamma (γ) = 0.09 for Data Set I and Data Set II, respectively (see Table 2).

3.3. Predictive Performance of Backpropagation and SVMs

Table 2 shows the predictive performance of the Backpropagation model with edge-detected images of road traffic scenes. The value of epoch, RMSE, average prediction error, variance and prediction accuracy in Table 2 is an average of 30 trials of the different initial weight vectors, since the Backpropagation model is sensitive to the initial weights. With edge-detected image of 30 pixels by 60 pixels, the average prediction errors are 98.63 in the test of a total of 700 data sets. On the other hand, with edge-detected images of 15 pixel by 30 pixel size, average prediction errors are 95.73 in the test of a total of 700 data sets. The predictive accuracies of two input vectors, edge-detected images of 30 pixels by 60 pixels and 15 pixels by 30 pixels, are 85.91% and 86.32%, respectively.

Table 3 shows the predictive performance of SVMs (Support Vector Machines) with edge-detected images of road traffic scenes. The predictive errors of SVMs (Support Vector Machines) on edge-detected images of 30 pixels by 60 pixels are 27, which is much lower than for the Backpropagation model. Also, the predictive errors of SVMs (Support Vector Machines) on edge-detected images of 15 pixels by 30 pixels were 20, even though there were 95.73 errors in Backpropagation model. On the other hand, the predictive accuracies of SVMs are 96.14% and 97.14% on edge-detected images of 30 pixels by 60 pixels and 15 pixels by 30 pixels, respectively. The results by experiments with edge-detected images show that SVMs (Support Vector Machines) could provide much better performance than the Backpropagation model. The best performance regarding recognition accuracy was 97.14% on the SVMs (Support Vector Machines) with Gamma (γ) = 0.02 and a 15 pixels by 30 pixels of image size. Figure 2 shows the comparison of predictive performance of two models, Backpropagation and Support Vector Machines.

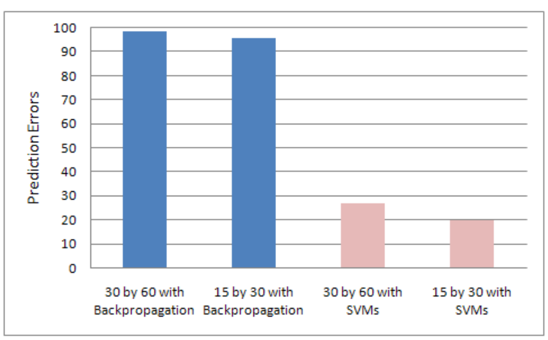
Table 2. Predictive performance of Backpropagation on Sobel edge-detection

Category	Image size(pxl)	
	30 by 60	15 by 30
Epoch	70.57	104.40
RMSE	0.001627	0.003159
Average Prediction Errors	98.63	95.73
Variance	1145.90	1477.31
Predictive accuracy (%)	85.91	86.32

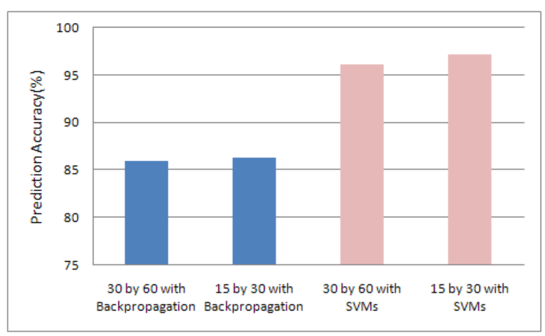
Source : Kim(2010)

Table 3. Predictive performance of SVMs on Sobel edge-detection

Category	Image size(pxl)	
	30 by 60	15 by 30
C	1	1
Gamma(γ)	0.02	0.09
Mean Squared Error	0.0771429	0.0457143
Prediction errors	27	20
Predictive accuracy (%)	96.14	97.143



(A) Prediction errors



(B) Prediction accuracy

Figure 2. Prediction errors and accuracy of two models

4. Conclusion

This paper has applied SVMs (Support Vector Machines) and showed their performance on edge-detected images for traffic scene analysis. The pattern recognition performance of SVMs was compared with that of Backpropagation neural network model. The study results based on several experiments show that SVMs can provide higher recognition performance in real-world traffic scenes analysis than Backpropagation, which is currently the most popular neural network model.

The SVMs may be more efficient and widely applicable than conventional methods, in classifying various complicated images with occlusion, shadow from other objects and noise problems, which factors are inevitable in real world images. Even though this paper showed that SVMs could provide higher prediction accuracy than the Backpropagation model, this result cannot be conclusive. The predictive accuracy of a model differs, according to the images and problems that it has to deal with. Although it seems likely that similar results would be achieved, more experiments are therefore needed to validate the study results.

Acknowledgements

This study was financially supported by Chonnam National University, 2010

References

[1] V. Vapnik, “The Nature of Statistical Learning Theory”, Springer Verlag, **(1995)**.
[2] V. Blanz, B. Schölkopf, H. Bülthoff, C. Burges, V. Vapnik and T. Vetter, “Comparison of view-based object recognition algorithms using realistic 3D models”, In: Artificial Neural Networks –ICANN’96, Berlin, Springer Lecture Notes in Computer Science, vol. 1112, **(1996)**, pp. 251-256.
[3] C. Cortes and V. Vapnik, “Support Vector Networks”, Machine Learning, vol. 20, **(1995)**, pp. 273-297.
[4] B. Schölkopf, C. Burges and V. Vapnik, “Extracting Support Data for a Given Task”, In: U. M. Fayyad and R. Uthurusamy(Eds.), Proceedings of 1st International Conference on Knowledge Discovery & Data Mining, Menlo Park, AAAI Press, **(1995)**.
[5] B. Schölkopf, C. Burges and V. Vapnik, “Incorporating Invariances in Support Vector Learning Machines”, In: C. von der Malsburg, W. von Seelen, J. C. Vorbrüggen and B. Sendhoff (Eds.), Artificial Neural Networks –ICANN’96, Berlin, Springer Lecture Notes in Computer Science, vol. 1112, **(1996)**, pp. 47-52.
[6] T. Joachims, “Text Categorization with Support Vector Machine: Learning with Many Relevant Features”, Proceedings of ECML-98, 10th European Conference on Machine Learning, **(1998)**, pp. 137-142.
[7] E. Osuna, R. Freund and F. Girosi, “Training Support Vector Machines: an Application to Face Detection”, IEEE Conference on Computer Vision and Pattern Recognition, **(1997)**, pp. 130-136.
[8] D. Kim, Y. Lim and S. Cho, “Performance improvement in traffic vision systems using SVMs”, Journal of the Eastern Asia Society for Transportation Studies, vol. 6, **(2005)**, pp. 2589-2599.
[9] D. E. Rumelhart, G. E. Hinton and J. L. McClelland, “A general framework for parallel distributed processing”, In D. E. Rumelhart, J. L. McClelland and the PDP Research Group (Eds). Parallel Distributed Processing, (Vol. 1 &2). Cambridge, Massachusetts: MIT Press, **(1986)**.
[10] S. E. Fahlman, “An empirical study of learning speed in backpropagation networks”, Technical Report CMU-CS-88-162, Canegie Mellon University, **(1988)**.
[11] D. Kim, “Standard and advanced backpropagation models for image processing application in traffic engineering”, Intelligent Transportation Systems, vol. 7, **(2002)**, pp. 199-211.
[12] Kim, D. “Prediction Performance of Support Vector Machines on Input Vector Normalization Methods”, International Journal of Computer Mathematics, vol. 81, no 5, **(2004)**, pp. 547-554.
[13] R. P. Lippmann, “An introduction to computing with neural nets”, IEEE Acoustics, Speech and Signal Processing Magazine, vol. 4, **(1987)**, pp. 4-22.
[14] A. Wieland and R. Leighton, “Geometric analysis of neural network Capabilities”, 1st IEEE International Conference on Neural Networks, vol. 3, **(1987)**, pp. 385-392.

[15] M. Arai, "Bounds on the number of hidden units in binary-valued three-layer neural networks", Neural Networks, vol. 6, **(1993)**, pp. 855-860.
[16] D. Kim, "Pre-processing of inputs to a neural network model for better performance in traffic scene analysis", Civil Engineering and Environmental Systems, vol. 27, no. 1, **(2010)**, pp. 23-31.