14233

OPSEC REVIEW CERTIFICATION

(AR 530-1, Operations Security)

I am aware that there is foreign intelligence interest in open source publications. I have sufficient technical expertise in the subject matter of this paper to make a determination that the net benefit of this public release outweighs any potential damage.

Reviewer: PAUL BOUNKER (Name) GS - 14 (Grade) COMPUTER ENGINEER (Title)

Paul Bounker (Signature) 7/2/01 (Date)

Description of Information Reviewed:

Title: A support vector machine application on vehicles

Author/Originator(s): Michael Del Rose

Publication/Presentation/Release Date: SPIE - 30-July-01

Purpose of Release: for publication and presentation

An abstract, summary, or copy of the information reviewed is available for review.

Reviewer's Determination (circle one):

1. Unclassified Unlimited.

2. Unclassified Limited, Dissemination Restrictions IAW ________.

3. Classified. Cannot be released, and requires classification and control at the level of ________.

Security Office (AMSTA-CS-S):

Concur/Nonconcur ________ (Signature) 2 JUL 01 (Date)

Public Affairs Office (AMSTA-CS-CT):

Concur/Nonconcur ________ (Signature) 3 July 01 (Date)

A support vector machine application on vehicles

Michael Del Rose, Jack Reed
U.S. Army TACOM Research and Development Center

Abstract

In this paper, methods of choosing a vehicle out of an image are explored. Digital images are taken from a monocular camera. Image processing techniques are applied to each single frame picture to create the feature vector. Finally the resulting features are used to classify whether there is a car in the picture or not using support vector machines. The result are compared to those obtained using a neural network. A discussion on techniques to enhance the feature vector and th results from both learning machines will be included.

Keywords: support vector machines, neural networks, image processing, pattern recognition, computer vision

1. Introduction

An obvious part to vehicle intelligence is having the ability to determine a car in front of you or not. This paper describes one algorithm to do just that. There are many other algorithms developed to do this. In ours, we use a multi-stage process to create the feature vector and support vector machines to classify it.

Support vector machines (SVM) are wide margin classifiers that solve a quadratic programming problem of maximum separation between two classes.[1-4] Developed by V. Vapnik, SVM most common uses are classification and regression problems with sparse data sets. In our project, an SVM classifier algorithm is applied to a feature vector. The feature vecto was processed from an image set of lanes with cars and empty lanes. The images were compiled from a monocular digital camera mounted on the dash of a car. The set is representative of common driving and weather conditions. The results from the SVM are then compared to a standard neural network.

Section two describes what support vector machines are and gives a very brief overview of how they work. The reader can skip this section if a basic knowledge of SVM is known. Section three describes the image processing techniques used to create our feature vector and the SVM pattern recognition procedure to get the results. Section four discusses the architecture of the neural network pattern recognition procedure. The results of both classifiers are found in section five and the conclusion is in section six.

2. Support Vector Machine Overview

2.1 Introduction

Support vector machine (SVM) theory is a learning machine theory developed by V. Vapnik. It is most commonly used fo classification and regression problems using small data sets. Like other learning machines, the distribution of the population does not need to be known. It is sufficient only to know that a distribution exists.[1-4]

The idea behind many classification problems is to find separating surfaces to divide each data class. SVM classifiers divide data between two classes. Generally, one of the class outputs is represented by 1 and the other by –1.

2.2 Support vector machine process

Figure 2.1 shows a linear separable set of data between two classes. Assume class 1 is represented by the set of data points in the upper right of the graph (the *s). Class –1 will be the data points in the lower left corner of the graph (the Xs).

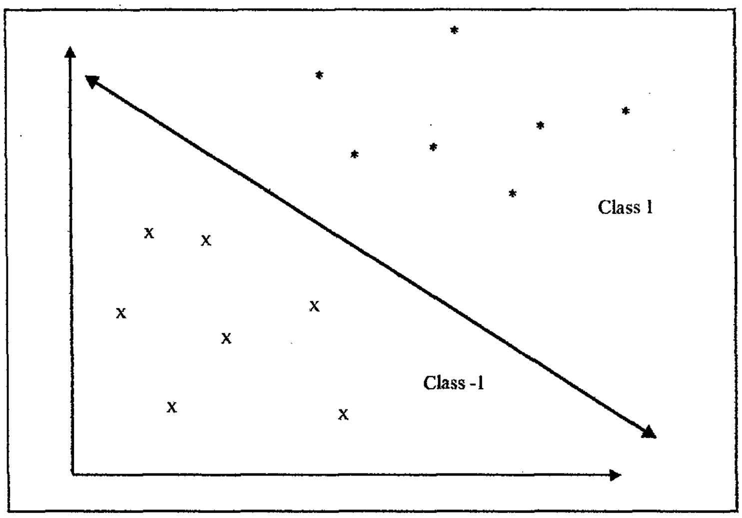

Figure 2.1

The goal of the SVM process for classification is to maximize the distance between the separating line and each data set. Figure 2.2 gives a graphical representation of what the SVM algorithm is doing.

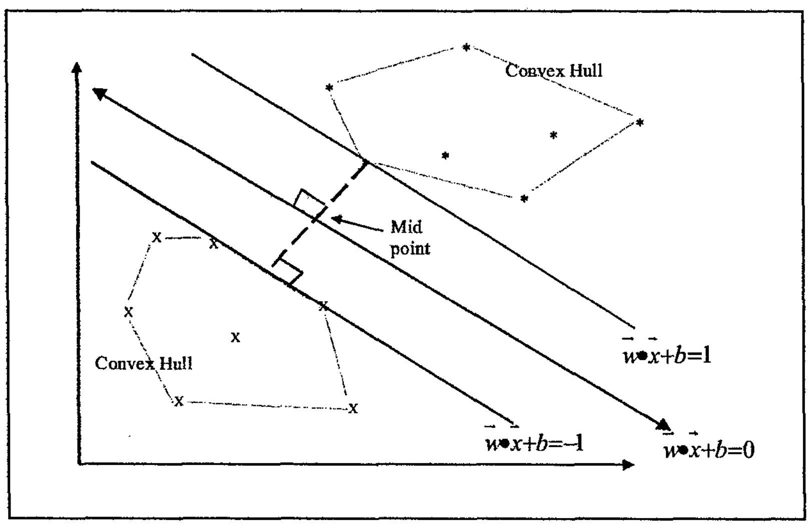

Figure 2.2

The steps of figure 2.2 are as follows:

1.) Draw a convex hull around each data set.
2.) Calculate the distance between each point in one class and each side of the convex hull in the opposing class.
3.) Choose the line, which makes up the smallest distance in part 2. This line will be called the 'Minimum Distance Line' or MDL.
4.) Calculate a line that intersects the midpoint of the MDL and is para llel to the side of the hull used for calculating the MDL. This line is the separating line.

Notice that there are three points (in this case) that determine the separating line: One from class 1 and two from class -1. These three points are considered the support vectors.

The solution to this problem becomes an optimization exercise. For the non-separable case it can be shown that changing th upper limit to the LaGrange multipliers will produce the desired results.

For a more analytical description of support vector machines, please consult the references.

3. Image processing and pattern recognition of the data set

3.1 Introduction

A monocular camera was mounted to the center of the dash board in a car. Digital images were retrieved of various driving and weather conditions. Pictures of cars in front and empty lanes were taken. Examples of typical pictures taken are shown in figure 3.1 and 3.2. Many different image-processing techniques were applied to the images. The following section gives a synopsis of the best pre-processing technique to get the desired result.

Figure 3.1
Typical image of a car

Figure 3.2
Typical image of a road without a car

3.2 Image pre-processing techniques

Many techniques exist to determine a car, a road, etc.[5-8] Each technique is beneficial in warning a driver of possible hazardous situations. The following pre-processing technique makes use of the assumption that cars produce a great number of straight lines compared to naturally occurring objects. In this technique, many parameters were varied. This paper onl describes the best method we found.

First, images of various driving conditions, which include cars in front of the camera and empty roads, were collected. The pictures were grayed and reduced to 400x400 using nearest neighbor so that all images would be of the same size. Since onl the middle portion of the picture was needed to determine if a car was in front of you, the pictures were trimmed to 250x250. This was done by cutting seventy-five pixels off the top, bottom, and each side. This removed most non-targeted cars (and noise).

The next step was to try to highlight the car from the rest of the image. A horizontal edge mask was used to get the horizontal edges from the image. Each line six pixels or greater was kept. The rest were regarded as noise and discarded. An image representation of this can be seen in figure 3.3.

We assumed that cars give off lots of horizontal lines, so the set of lines which represent a car should be close together. So, we removed any line that was not within 20 pixels of another line. This reduced a lot of the noise and highlighted the ca from the rest of the environment.

Out last step was to group the lines into lengths. The reason for this was to reduce the feature vector. A histogram was made to encompass the following line pixel length groups[8,9]: <11, 11-13, 14-16, 17-19, 20-22, 22<. One final feature of the total number of lines left was added to the feature vector. The feature vector consisted of seven elements per image.

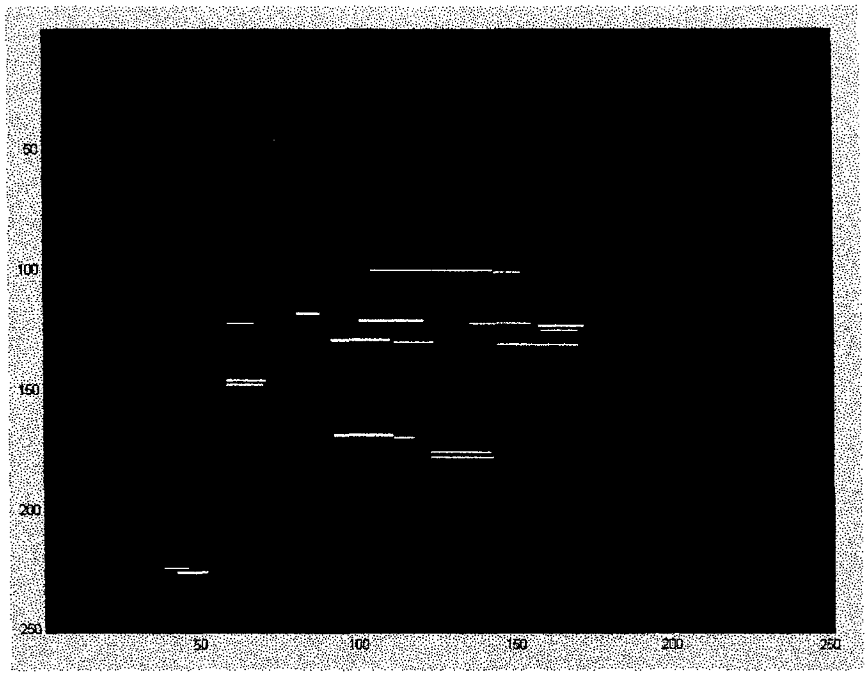

Figure 3.3
Horizontal lines of image 3.1 six pixels or greater

3.3 Pattern recognition using support vector machines
A support vector machine was used to train and test the feature vector. 267 images were used. Of these images, a random sample of twenty percent of car and no car situations were used to train the SVM. The remaining images were used to test the SVM classifier. A quadratic polynomial kernel was used for the SVM. The SVM code used was from Steve Gunn's Matlab toolbox. The results are found in section five.

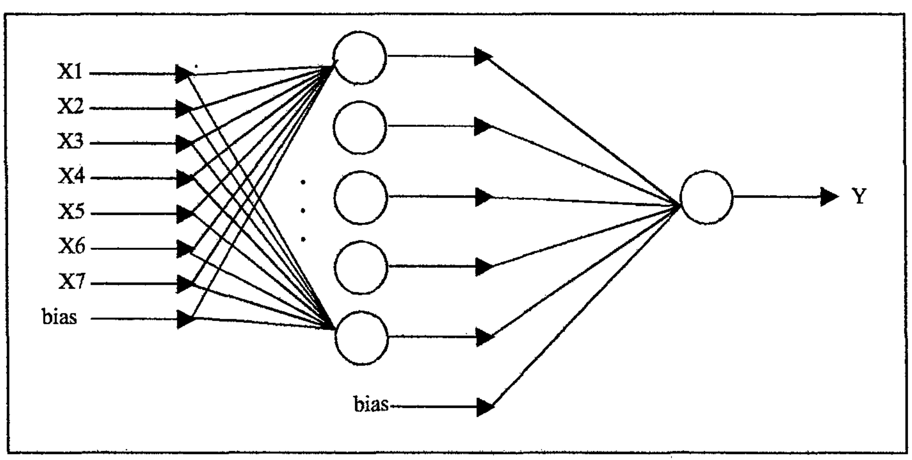

Figure 4.1
Diagram of neural network used in this classification problem

4. Neural Network Approach

A standard neural network was used to test the results from the support vector machine classifier. The same test data and training data was used. The neural network classifier used a multi-layer architecture with a unipolar sigmoid activation function for each of its neuron [10]. The hidden layer had five neurons and the outer layer had one neuron (since the output is binary). A diagram of the neural network architecture is shown in figure 4.1.

5. Results

The support vector machine classifier accurately classified 100 percent of the images using the above criteria for the feature vector. The neural network only classified about 89 percent with 98 percent correctly classifying car images and only 81 percent correctly classifying no car images. When the feature vectors changed, such as the minimum line pixel length changed, the SVM did not do as well, but it still out performed the neural network.

6. Conclusion

It was expected that the support vector machine would out perform the neural network since we are working with a small data set. Although we tried to capture many driving and weather conditions, we could not get them all. Adding more data into the system may prove that neural network classifiers will outperform SVM classifiers. Future work should include a much larger data set. Also, the SVM classifier algorithm discussed should be compared against other, more accepted, algorithms.

7. Acknowledgement

We would like to thank the National Automotive Center (NAC) for providing funds to continue the research on this project. We would also like to thank Dr. Dave Gorsich for his guidance on this project.

8. References

1. B. Scholkopf, C. Burges, A. Smola, *Advances in Kernel Methods*, MIT Press, Cambridge, 1999.
2. V. Vapnik, *Statistical Learning Theory*, John Wiley and Sons, New York, 1998.
3. S. Gunn, *Support Vector Machines for Classification and Regression*, Technical Report, Image Speech and Intelligent Systems Group, Univ. of Southhampton, 1998.
4. R. Karlsen, D. Gorsich, G. Gerhart, "Target Classification Via Support Vector Machines," *Optical Engineering*, **39**:704-711, 2000.
5. H. Burdick, *Digital Imaging, Theory and Application*, McGraw-Hill, New York, 1997.
6. G. Baxes, *Digital Image Processing, Principles and Applications*, John Wiley and Sons, New York, 1994.
7. R. Crane, *A Simplified Approach to Image Processing*, Prentice Hall, Upper Saddle River, 1997.
8. M. Del Rose, *Image Processing Methods*, Technical Report, University Of Michigan, 1999.
9. M. Del Rose, *Hand Written Character Recognition*, Masters Thesis, University of Michigan, 2000.
10. C. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, New York, 1995.