A SUPPORT VECTOR MACHINE APPROACH FOR OBJECT BASED IMAGE ANALYSIS

Angelos Tzotsos*

Laboratory of Remote Sensing, Department of Surveying, School of Rural and Surveying Engineering, National Technical University of Athens, Greece – tzotsos@gmail.com

Commission IV, WG IV/4 – Commission VIII, WG VIII/11

KEY WORDS: Remote Sensing, Computational Intelligence, Machine Learning, Object Oriented Classification, Multi-scale Segmentation.

ABSTRACT:

The Support Vector Machine is a theoretically superior machine learning methodology with great results in classification of high-dimensional datasets and has been found competitive with the best machine learning algorithms. In the past, SVMs have been tested and evaluated only as pixel-based image classifiers. Moving from pixel-based techniques towards object-based representation, the dimensions of remote sensing imagery feature space increases significantly. This results increasing complexity of the classification process, and causes problems to traditional sample-based classification schemes. The objective of this study was to evaluate SVMs for effectiveness and prospects for object-based image classification as a modern computational intelligence method. An SVM approach for multi-class classification was followed, based on primitive image objects produces by a multi-resolution segmentation algorithm. The segmentation algorithm produced primitive objects of variable sizes and shapes. Then, a feature selection step took place in order to provide the features for classification which involved spectral, texture and shape information. Contextual information was not used. Following the feature selection step, a module integrating an SVM classifier and the segmentation algorithm was developed in C++ and based on XML technology for feature representation. For training the SVM, sample image objects, derived from the segmentation procedure were used. The SVM procedure produced the final object classification results which were compared to the Nearest Neighbor classifier results, of the eCognition software, and were found satisfactory. The SVM approach seems very promising for Object Based Image Analysis and future work will focus on the integration SVM classifiers with rule-based classifiers.

1. INTRODUCTION

1.1 Knowledge-based image classification and Object Oriented Image Analysis

In recent years, research has progressed in computer vision methods applied to remotely sensed images such as segmentation, object oriented and knowledge-based methods for classification of high-resolution imagery (Argialas and Harlow 1990, Kanellopoulos et al. 1997). In Computer Vision, image analysis is considered in three levels: low, medium and high (Argialas and Harlow 1990). Such approaches were usually implemented in separate software environments since low and medium level algorithms are procedural in nature, while high level is inferential and thus for the first one needs procedural languages while for the second an expert system environment is more appropriate.

New approaches have been developed, recently in the field of Remote Sensing. Some of them were based on knowledge-based techniques in order to take advantage of the expert knowledge derived from human photo-interpreters (Argialas and Goudoula 2003, Yoo et al 2002, Yooa et al 2005). In particular within an Expert System environment, the classification step has been implemented through logic rules and heuristics, operating on classes and features, which were implemented by the user through an object-oriented representation (De Moraes 2004, Moller-Jensen 1997). This object-oriented representation was mainly based on the image semantics and the explicit knowledge of the human expert. In order to classify each element of the image into the appropriate class, the knowledge based expert system represented the definitions of the classes through rules and heuristics, which an expert explicitly declares and develops within the system. As a result, more complex methods for image classification have been implemented and many more image features can be used for the classification step (Smits and Annoni 1999, Kanellopoulos et al. 1997).

Very recently a new methodology called Object Oriented Image Analysis was introduced, integrating low-level, knowledge-free segmentation with high-level, knowledge-based fuzzy classification methods. This new methodology was implemented through a commercial software, eCognition, which incorporated an object-oriented environment for the classification of satellite imagery (Baatz and Shape 2000, Benz et al. 2004).

1.2 Computational Intelligence methods in Remote Sensing

Other fields of Artificial Intelligence have also been developed such as Computational Intelligence and Machine Learning involving Neural Networks, Fuzzy Systems, Genetic Algorithms, Intelligent Agents and Support Vector Machines (Negnevitsky 2005). Machine learning is an integral part of Pattern Recognition, and in particular classification (Theodoridis and Koutroumbas 2003). Given that in the past, digital remote sensing used pattern recognition techniques for

* Heroon Polytechneiou 9, 15780, Zografou, Athens, Greece. Tel. +302107722684

classification purposes, modern machine learning techniques have been also implemented for remote sensing applications and achieved very good classification results (Binaghi et al 2003, Fang and Liang 2003, Theodoridis and Koutroumbas 2003, Huang et al 2002, Brown et al 2000, Foody and Mathur 2004).

The Support Vector Machine (SVM) is a theoretically superior machine learning methodology with great results in the classification of high-dimensional datasets and has been found competitive with the best machine learning algorithms. In the past, SVMs were tested and evaluated only as pixel based image classifiers with very good results (Huang et al 2002, Brown et al 2000, Foody and Mathur 2004, Gualtieri and Cromp 1999, Melgani and Bruzzone 2004).

Furthermore, for remote sensing data it has been shown that Support Vector Machines have great potential, especially for hyperspectral data, due to their high-dimensionality (Gualtieri and Cromp 1999, Melgani and Bruzzone 2004). In recent studies, Support Vector Machines were compared to other classification methods, such as Neural Networks, Nearest Neighbor, Maximum Likelihood and Decision Tree classifiers for remote sensing imagery and have surpassed all of them in robustness and accuracy (Huang et al 2002, Foody and Mathur 2004).

1.3 Research Objectives

The objective of this study was to evaluate SVMs for their effectiveness and prospects for object-based image classification.

A secondary objective was to evaluate the accuracy of SVM compared to simpler and widely used classification techniques such as Nearest Neighbor. Also, the computational efficiency and training size requirements of SVMs were set for consideration.

2. METHODOLOGY

2.1 Multi-scale Segmentation

Image segmentation is an integral part of Object-Based Image Analysis methodology (Benz et al 2004). The digital image is no longer considered as a grid of pixels, but as a group of primitives and homogeneous regions, called primitive image objects. The object oriented representation provides to the classification process information that could not be derived from single pixels such as context and shape information. These are very important factors to photo-interpretation and image understanding (Lillesand and Kiefer 1987, Sonka et al 1998, Biederman 1985). Objects can be more intelligent than pixels, in a sense of knowing their "neighbours" and the spatial or spectral relations with and among them.

In order to perform object based classification, a segmentation algorithm is needed to provide knowledge-free primitive image objects. When a photo interpretation task is carried out by an expert, the scale of imagery is specified by the nature of image semantics to be recognized (Lillesand and Kiefer 1987). During the higher level image classification steps, there is a need to have primitive objects of different sizes and preferably on different scales of abstraction derived from the sameimagery (Tzotsos and Argialas 2006, Baatz and Shape 2000, Benz et al. 2004). That is the main reason why for remote sensing image classification, a multi-resolution segmentation approach is needed.

For this research effort the MSEG multi-scale segmentation algorithm was used (Tzotsos and Argialas 2006). The main reason for this choice was that it has an open architecture to implement new features in C++. For evaluation purposes, the Multiresolution Segmentation algorithm in eCognition was also used (Baatz and Shape 2000).

MSEG can be described as a region merging procedure. The first primitive object representation is the single image pixel. Through iterative pairwise object fusions, which are made at several iterations called passes, the final segmentation is achieved. The criterion for object merging is a homogeneity cost measure, defined as object heterogeneity, and computed based on spectral, texture and shape features for each possible object merge. The heterogeneity is then compared to a user defined threshold, called scale parameter, to determine the decision of the merge. MSEG also offers a multi-resolution algorithm which performs segmentations at several levels and at the same time provides automatic topology of objects within each level and among levels (Tzotsos and Argialas 2006).

2.2 Support Vector Machines

Recently, particular attention has been dedicated to Support Vector Machines as a classification method. SVMs have often been found to provide better classification results that other widely used pattern recognition methods, such as the maximum likelihood and neural network classifiers (Melgani and Bruzzone 2004, Theodoridis and Koutroumbas 2003). Thus, SVMs are very attractive for the classification of remotely sensed data.

The SVM approach seeks to find the optimal separating hyperplane between classes by focusing on the training cases that are placed at the edge of the class descriptors. These training cases are called support vectors. Training cases other than support vectors are discarded. This way, not only is an optimal hyperplane fitted, but also less training samples are effectively used; thus high classification accuracy is achieved with small training sets (Mercier and Lennon 2003). This feature is very advantageous, especially for remote sensing datasets and more specifically for Object-based Image Analysis, where object samples tend to be less in number than in pixel-based approaches.

A complete formulation of Support Vector Machines can be found at a number of publications (Vapnik 1995, 1998, Cortes and Vapnik 1995, Theodoridis and Koutroumbas 2003). Here, the basic principles will be presented and then their implementation and application to Object Based Image Analysis will be evaluated.

Let us consider a supervised binary classification problem. If the training data are represented by $\{x_i, y_i\}$, i = 1, 2, …, N, and $y_i \in \{-1, +1\}$, where N is the number of training samples, $y_i=+1$ for class ω_1 and $y_i=-1$ for class ω_2. Suppose the two classes are linearly separable. This means that it is possible to find at least one hyperplane defined by a vector w with a bias w_0, which can separate the classes without error:

$$f(x) = w \cdot x + w_0 = 0 \qquad (1)$$

To find such a hyperplane, w and w_0 should be estimated in a way that $y_i(w \cdot x_i + w_0) \geq +1$ for $y_i = +1$ (class ω_1) and $y_i(w \cdot x_i + w_0) \leq -1$ for $y_i = -1$ (class ω_2). These two, can be combined to provide equation 2:

$$y_i(w \cdot x_i + w_0) - 1 \geq 0 \qquad (2)$$

Many hyperplanes could be fitted to separate the two classes but there is only one optimal hyperplane that is expected to generalize better than other hyperplanes (Figure 1).

The goal is to search for the hyperplane that leaves the maximum margin between classes. To be able to find the optimal hyperplane, the support vectors must be defined. The support vectors lie on two hyperplanes which are parallel to the optimal and are given by:

$$w \cdot x_i + w_0 = \pm 1 \quad (3)$$

If a simple rescale of the hyperplane parameters w and w_0 takes place, the margin can be expressed as $\frac{2}{\|w\|}$. The optimal hyperplane can be found by solving the following optimization problem:

Minimize $\frac{1}{2}\|w\|^2$ (4)

Subject to $y_i(w \cdot x_i + w_0) - 1 \geq 0$ i = 0, 1, … N

Using a Lagrangian formulation, the above problem can be translated to:

Maximize $\sum_{i=1}^{N} \lambda_i - \frac{1}{2}\sum_{i,j=1}^{N} \lambda_i \lambda_j y_i y_j (x_i \cdot x_j)$ (5)

Subject to $\sum_{i=1}^{N} \lambda_i y_i = 0$ and $\lambda_i \geq 0$, i = 1, 2, …N

where λ_i are the Lagrange multipliers.

Under this formulation, the optimal hyperplane discriminant function becomes:

$$f(x) = \sum_{i \in S} \lambda_i y_i (x_i x) + w_0 \quad (6)$$

where S is a subset of training samples that correspond to non-zero Lagrange multipliers. These training samples are called support vectors.

In most cases, classes are not linearly separable, and the constrain of equation 2 cannot be satisfied. In order to handle such cases, a cost function can be formulated to combine maximization of margin and minimization of error criteria, using a set of variables called slack variables ξ (Figure 1). This cost function is defined as:

Minimize $J(w, w_0, \xi) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} \xi_i$ (7)

Subject to $y_i(w \cdot x + w_0) \geq 1 - \xi_i$

To generalize the above method to non-linear discriminant functions, the Support Vector Machine maps the input vector x into a high-dimensional feature space and then constructs the optimal separating hyperplane in that space. One would consider that mapping into a high dimensional feature space would add extra complexity to the problem. But, according to the Mercer's theorem (Vapnik 1998, Theodoridis and Koutroumbas 2003), the inner product of the vectors in the mapping space, can be expressed as a function of the inner products of the corresponding vectors in the original space.

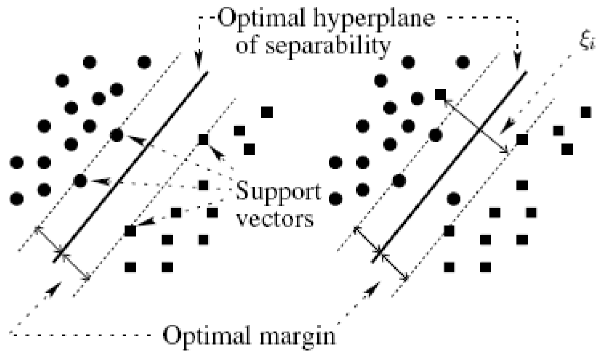

Figure 1: Left: The case of linear separable classes. Right: The case of non linear separable classes. ξ measures the error of the hyperplane fitting. (source: Mercier and Lennon 2003)

The inner product operation has an equivalent representation:

$$\Phi(x)\Phi(z) = \mathrm{K}(x, z) \quad (8)$$

where K(x,z) is called a kernel function. If a kernel function K can be found, this function can be used for training without knowing the explicit form of Φ.

The dual optimization problem is now formed as:

Maximize $\sum_{i=1}^{N} \lambda_i - \frac{1}{2}\sum_{i,j=1}^{N} \lambda_i \lambda_j y_i y_j \mathrm{K}(x_i \cdot x_j)$ (9)

Subject to: $\sum_{i=1}^{N} \lambda_i y_i = 0$ and $\lambda_i \geq 0$, i = 1, 2, …N

The resulting classifier becomes:

$$f(x) = \sum_{i \in S} \lambda_i y_i \mathrm{K}(x_i x) + w_0 \quad (10)$$

2.3 SVM Multi-class Classification

The SVM method was designed to be applied only for two class problems. For applying SVM to multi-class classifications, two main approaches have been suggested. The basic idea is to reduce the multi-class to a set of binary problems so that the SVM approach can be used.

The first approach is called "one against all". In this approach, a set of binary classifiers is trained to be able to separate each class from all others. Then each data object is classified to the class for which the largest decision value was determined (Hsu and Lin 2002). This method trains N SVMs (where N is the number of classes) and there are N decision functions. Although it is a fast method, it suffers from errors caused by marginally imbalanced training sets. Another approach was recently introduced (Hsu and Lin 2002), which is similar to the "one against all" method, but uses one optimization problem to obtain the N decision functions (equation 10). Reducing the classification to one optimization problem may require less support vectors than a multi-class classification based on many binary SVMs.

The second approach is called "one against one". In this, a series of classifiers is applied to each pair of classes, with the most commonly computed class kept for each object. Then a max-win operator is used to determine to which class the object will be finally assigned. The application of this method requires N(N-1)/2 machines to be applied. Even if this method is more computationally demanding than the "one against all" method, it has been shown that it can be more suitable for multi-class classification problems (Hsu and Lin 2002), thus it was selected for SVM object-based image classification.

2.4 Implementation

In order to apply the SVM methodology for Object-Based Image Analysis, it was necessary to perform a segmentation of the image. The MSEG algorithm was selected to perform segmentation at multiple scales (Tzotsos and Argialas 2006) and to produce primitive image objects to be used for SVM classification.
For the primitive objects, to be usable by a classification algorithm, there was a need to implement an interface between image objects and the classifier. This interface should include an object feature export mechanism and also a way to provide training data for the classifier.
An extra module was implemented into the MSEG core library to add the functionality of selecting sample objects. Because a comparison was to be made with the Nearest Neighbor classifier used in eCognition, a TTA Mask (eCognition user guide 2005) import module was also implemented, so that the training object selection process would be as transparent and objective as possible.
For the object feature interface, the XML language was selected, so that open standards are followed. An XML representation was implemented for the segmentation level class, to provide the classifier all the information about the segmentation procedure that was performed to produce the object primitives. In Figure 3, a typical XML level file is presented.
A widely used SVM library called LIBSVM (Chang and Lin 2001) was then modified to be able to handle XML level files as well as training samples from the MSEG algorithm. A classifier module was then implemented as a modified version of LIBSVM.
The proposed Object-based Image Analysis system worked in the following way: A segmentation procedure was carried out with scale, color and shape parameters. The properties of the primitive objects were then computed and exported to XML format (Figure 2). A TTA Mask file along with its attribute table was imported to the system and training object samples were defined. A training set of feature vectors was exported from the MSEG algorithm and was used for training the SVM module.
The SVM module is capable of using 4 types of kernels for training and classification:

Linear: $$\mathrm{K}(x_i x_j) = x_i^T x_j$$

Polynomial: $$\mathrm{K}(x_i x_j) = (\gamma \cdot x_i^T x_j + r)^d, \gamma > 0$$

Radial Basis Function (RBF):

$$\mathrm{K}(x_i x_j) = \exp(-\gamma \cdot \left\| x_i - x_j \right\|^2), \gamma > 0$$

Sigmoid: $$\mathrm{K}(x_i x_j) = \tanh(\gamma \cdot x_i^T x_j + r)$$

where γ, r and d are kernel parameters.

All the above kernels follow Mercer's theorem and can be used for mapping the feature space into a higher dimensional space to find an optimal separating hyperplane. In literature, there have been many comparison studies between the most common kernels (Mercier and Lennon 2003, Huang et al 2002). For pixel-based classification of remotely sensed data, it has been shown that local kernels such as RBF can be very effective and accurate. Also, the linear kernel is a special case of the RBF kernel, with specific parameters (Hsu and Lin 2002). Based on the above, for the current study only RBF kernels were used.
For the training of the SVM classifier, the error parameter C (equation 7) and the kernel parameter γ had to be obtained. In order to find the optimal parameters for the RBF kernel function a cross-validation procedure was followed.


```xml
<?xml version="1.0" ?>
<Level>
    <LevelID type="float">1</LevelID>
    <RasterFilename>stats</RasterFilename>
    <Lines type="int">500</Lines>
    <Columns type="int">500</Columns>
    <Parameters>
        <ScaleParameter type="float">100</ScaleParameter>
        <Color type="float">0.8</Color>
        <Compactness type="float">0.4</Compactness>
        <Edge type="float">0</Edge>
    </Parameters>
    <Raster />
    <Properties>
        <Object>
            <id>471014</id>
            <area>62</area>
            <perimeter>46</perimeter>
            <Band>
                <BandID>Blue</BandID>
                <mean type="float">227.081</mean>
                <std type="float">9.13741</std>
            </Band>
            <Band>
                <BandID>Green</BandID>
                <mean type="float">223.258</mean>
                <std type="float">9.10158</std>
            </Band>
            <Band>
                <BandID>Red</BandID>
                <mean type="float">230.71</mean>
                <std type="float">6.54114</std>
            </Band>
```


Figure 2: XML representation of a segmentation level. Image object properties are included to be used by a classifier. Here only few (mean band value and standard deviation) of the available properties were used.

First the training set was scaled to the range of [-1, +1] to avoid features in greater numerical ranges dominating those in smaller ranges (Negnevitsky 2005). Then, the training set was divided to many smaller sets of equal size. Sequentially each subset was tested using the classifier trained by the remaining subsets. This way each image object is predicted once during the above process. The overall accuracy of the cross-validation is the percentage of correctly classified image objects.
After the cross-validation delivered the optimal parameters for the SVM classifier, the training set was used to train the SVM. Then the classifier was supplied with all image primitive objects so to derive the final object based classification. The output of the above procedure was a classification map as well as an updated XML representation of the segmentation level.

3. DISCUSSION OF RESULTS

For the evaluation of the developed approach, a Landsat TM image was used. For comparison purposes, an object-based classification of the same image was performed in eCognition. The training samples in both cases were the same (a TTA mask file) and were obtained by the eCognition user guide (2005) for objective evaluation. The original Landsat TM image and the training samples are presented in Figure 3. A reference dataset was also derived by photo-interpretation and was used to compute confusion matrices (Figure 4).

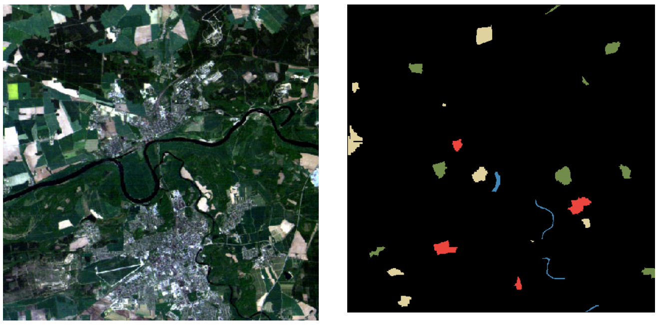

Figure 3: Left: the original Landsat TM image (source: eCognition User Guide 2005). Right: The training set of class samples (blue=Water, red=Impervious, green=Woodland and yellow=Grassland).

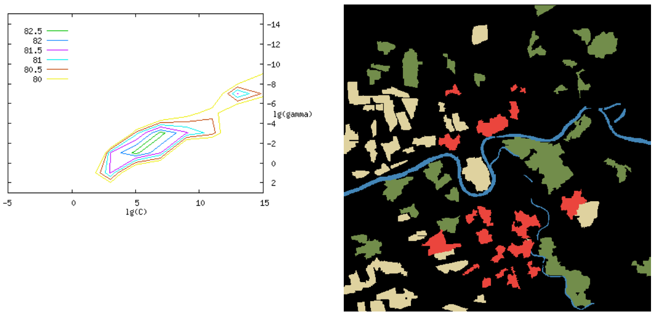

Figure 4: Left: The cross-validation plot diagram for selecting the optimal values of C and γ for SVM training. Right: The ground-truth dataset used to evaluate results

First, the training samples were projected upon small primitive objects that were derived by eCognition with scale parameter 10 and by MSEG with scale parameter 100. Scale parameters are compatible between these segmentation algorithms as they are identical internally (Tzotsos and Argialas 2006, eCognition user guide 2005). For the export of training samples, the minimum overlap for each sample object was set to 50%. The overall accuracy of the Nearest Neighbor (NN) method, based on the reference dataset was 85.6%. A cross-validation procedure was followed to provide the best C and γ parameters for the SVM classifier. The results of cross-validation are shown in Figure 4. The overall accuracy of the object-based SVM classification was 90.6% (Figure 5, Tables 1 and 2).

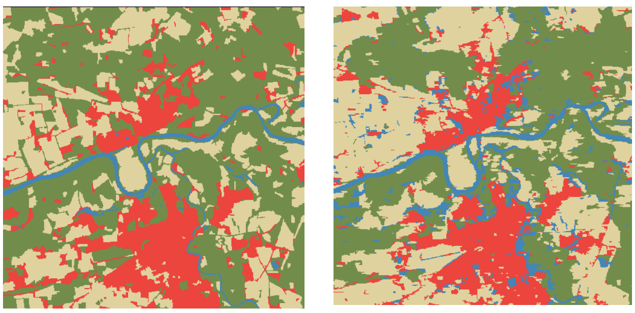

Figure 5: Left: eCognition classification result with Nearest Neighbor. Right: MSEG classification result with SVM. Training sample overlap with objects set to 50%.

	Woodland	Grassland	Impervious	Water
Woodland	17922	3381	280	0
Grassland	2578	12854	195	0
Impervious	139	770	8539	0
Water	80	0	0	4740

Table 1: Nearest Neighbor confusion matrix. The overall accuracy was 85.6%

	Woodland	Grassland	Impervious	Water
Woodland	17846	2088	45	740
Grassland	767	15937	210	91
Impervious	231	215	8305	263
Water	180	13	10	4537

Table 2: SVM confusion matrix. The overall accuracy was 90.6%

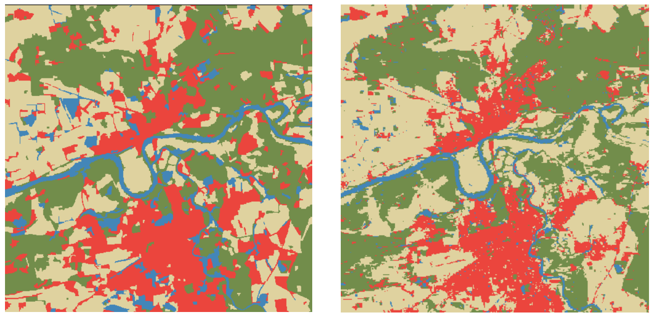

Figure 6: Left: eCognition classification result with Nearest Neighbor. Right: MSEG classification result with SVM. In both classifications, errors have been introduced to the training sets for generalization evaluation.

	Woodland	Grassland	Impervious	Water
Woodland	16080	1470	0	0
Grassland	2195	13891	195	0
Impervious	899	314	8605	0
Water	1545	1330	214	4740

Table 3: Nearest Neighbor confusion matrix. The overall accuracy was 84.1%

Then, in order to test the generalization ability of both classifiers, an error was introduced into the training samples, in the form of not using a minimum overlap restriction for sample object selection. This way, more training objects were selected with errors derived from the segmentation procedures. An interesting observation was that the SVM behaved better than the NN to the second training set and provided better classification results (Tables 3 and 4) giving an overall accuracy of 86.0% against 84.1% for the NN. Both classification results are presented in Figure 6.

	Woodland	Grassland	Impervious	Water
Woodland	16816	3458	207	238
Grassland	1262	15506	178	59
Impervious	249	325	8315	125
Water	349	755	1	3635

Table 4: SVM confusion matrix. The overall accuracy was 86.0%

4. CONCLUSIONS AND FUTURE WORK

Overall, the SVM classification approach was found very promising for Object-Based Image Analysis. It has been shown that it can produce comparable or even better results than the Nearest Neighbor for supervised classification.

The computational efficiency of SVM was great, with only a few minutes of runtime necessary for training. This was theoretically predicted but also, the implementation in C++ is extremely fast. However, very large remote sensing datasets were not tested.

A very good feature of SVMs is that only a small training set is needed to provide very good results, because only the support vectors are of importance during training.

Future work will include comparison of many SVM kernels for Object oriented image classification. Also, an integration of SVM classifiers with rule-based classifiers will be implemented for context-based classification.

5. ACKNOWLEDGEMENTS

The author would like to thank Erin for her support during this paper effort.

The author would also like to thank Professor Demetre Argialas for the corrections and suggestions he provided.

6. REFERENCES

Argialas D. and V.Goudoula 2003. Knowledge-based Land Use Classification from IKONOS Imagery for Arkadi, Crete, Greece. Remote Sensing for Environmental Monitoring, GIS Applications, and Geology II, Proceedings of SPIE Vol. 4886.

Argialas, D., and C. Harlow 1990. Computational Image Interpretation Models: An Overview and a Perspective. *Photogrammetric Engineering and Remote Sensing*, Vol. 56, No 6, June, pp. 871-886.

Baatz, M. & Schäpe, A. 2000. Multiresolution Segmentation – an optimization approach for high quality multi-scale image segmentation. *In: Strobl, J. et al. (eds.): Angewandte Geographische Infor-mationsverarbeitung XII. Wichmann, Heidelberg, pp. 12-23*.

Benz U., Hoffman P., Willhauck G., Lingenfelder I. and M. Heynen, 2004. Multi-resolution, object-oriented fuzzy analysis of remote sensing data for GIS-ready information. *ISPRS Journal of Photogrammetry and Remote Sensing 58* pp. 239-258.

Biederman, I. 1985. Human image understanding: Recent research and a theory. Computer Vision, Graphics, and Image Processing, 32, 29–73.

Binaghi E., Gallo I. and M. Pepe 2003. A Neural Adaptive Model for Feature Extraction and Recognition in High Resolution Remote Sensing Imagery Int. J. Remote Sensing, 2003, Vol. 24, No. 20, 3947–3959

Brown M., Lewis H.G., and S.R. Gunn, 2000. Linear Spectral Mixture Models and Support Vector Machines for Remote Sensing. IEEE Transactions On Geoscience And Remote Sensing, Vol. 38, No. 5, September 2000

Chang, C.-C. and C.-J. Lin, 2001. LIBSVM: a library for support vector machines. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm. (accessed 15/06/2006)

Cortes C. and V. Vapnik. Support-vector network. Machine Learning, 20:273–297, 1995.

De Moraes R.M. 2004. An Analysis Of The Fuzzy Expert Systems Architecture For Multispectral Image Classification Using Mathematical Morphology Operators (Invited Paper) International Journal of Computational Cognition (http://www.YangSky.com/yangijcc.htm) Volume 2, Number 2, Pages 35–69, June 2004.

eCognition User Guide, 2005, Definiens, Munchen. http://www.definiens.com (accessed 15/06/2006)

Fang H. and S. Liang 2003. Retrieving Leaf Area Index With a Neural Network Method: Simulation and Validation IEEE Transactions On Geoscience And Remote Sensing, Vol. 41, No. 9, September 2003.

Foody G.M. and A. Mathur, 2004. A Relative Evaluation of Multiclass Image Classification by Support Vector Machines. IEEE Transactions On Geoscience And Remote Sensing, Vol. 42, No. 6, June 2004.

Gualtieri J.A. and R. F. Cromp, 1999. Support vector machines for hyperspectral remote sensing classification," in *Proceedings of the SPIE*, vol. 3584, 1999, pp. 221–232.

Hsu C.-W. and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Trans. Neural Networks*, vol. 13, pp. 415–425, Mar. 2002.

Huang C., L. S. Davis, and J. R. G. Townshend 2002, An assessement of support vector machines for land cover classifiocation," *Int. J. Remote sensing*, vol. 23, no. 4, pp. 725–749, 2002.

Kanellopoulos I., Wilkinson G., and T. Moons (1999). *Machine Vision and Advanced Image Processing in Remote Sensing*. Springer Verlag.

Lillesand, T.M., Kiefer, R.W., 1987. Remote-Sensing and Image Interpretation. Wiley, New York.

Melgani F. and L. Bruzzone 2004. Classification of Hyperspectral Remote Sensing Images With Support Vector Machines. IEEE Transactions On Geoscience And Remote Sensing, Vol. 42, No. 8, August 2004.

Mercier G. and M. Lennon 2003. Support vector machines for hyperspectral image classification with spectral-based kernels. in *Proc. IGARSS*, Toulouse, France, July 21–25, 2003.

Moller-Jensen L. 1997. Classification of Ubrban Land Cover Based on Expert Systems, Object Models and Texture. Comput. Environ and Urban Systems, Vol.21, No. 3/4, pp. 291-302, 1997.

Negnevitsky M., 2005. Artificial Intelligence, a Guide to Intelligent Systems. Pearson Education, p.440, 2005.

Smits P.C., A. Annoni 1999. Towards operational knowledge-based remote-sensing image analysis. Pattern Recognition Letters 20 (1999) 1415 1422

Sonka, M., Hlavac, V. Boyle, R., 1998. *Image Processing, Analysis, and Machine Vision* - 2nd Edition, PWS, Pacific Grove, CA, 800 p., ISBN 0-534-95393-X.

Theodoridis S. and K. Koutroumbas 2003. Pattern Recognition. Second Edition. Elsevier Academic Press, 2003.

Tzotsos A. and D. Argialas, 2006. MSEG: A generic region-based multi-scale image segmentation algorithm for remote sensing imagery. Proceedings of ASPRS 2006 Annual Conference, Reno, Nevada; May 1-5, 2006.

Vapnik V.N. 1998. Statistical Learning Theory. John-Wiley and Sons, Inc.

Vapnik, V.N. 1995. The Nature of Statistical Learning Theory. New York, NY: Springer-Verlag.

Yoo, H. W., Jang, D. S., Jung, S. H., Park, J. H., and Song, K. S. 2002. Visual information retrieval system via content-based approach. Pattern Recognition, 35(3), 749–769.

Yooa H.W., H.S. Park, D.S. Jang 2005. Expert system for color image retrieval. Expert Systems with Applications 28 (2005) 347–357