A Particle Swarm Optimization-based Approach for Hyperspectral Band Selection

Sildomar Takahashi Monteiro, *Member, IEEE*, and Yukio Kosugi, *Member, IEEE*

***Abstract*— In this paper, a feature selection algorithm based on particle swarm optimization for processing remotely acquired hyperspectral data is presented. Since particle swarm optimization was originally developed to search only continuous spaces, it could not deal with the problem of spectral band selection directly. We propose a method utilizing two swarms of particles in order to optimize simultaneously a desired performance criterion and the number of selected features. The candidate feature sets were evaluated on a regression problem using artificial neural networks to construct nonlinear models of chemical concentration of glucose in soybean crops. Experimental results attesting the viability of the method utilizing real-world hyperspectral data are presented. The particle swarm optimization-based approach presented superior performance in comparison with a conventional feature extraction method.**

I. Introduction

Particle swarm optimization (PSO) is an evolutionary computation technique that has been developed due to research on bird flock simulation by Kennedy and Eberhart [1]. PSO is able to solve most optimization problems, or problems that can be converted to optimization problems. PSO's main attractiveness is its simplicity and velocity, allied with robustness.

Hyperspectral imaging sensors are able to acquire several hundreds of spectral information from the visible to the infrared region. Nonetheless, neighboring spectral bands are usually highly redundant [2]. In real-world applications, the typical scenario of few data samples in a high-dimensional feature space causes what was termed by Bellman [3] as the curse of dimensionality, referring to the exponential increase in complexity of high-dimensional spaces with the increase in the number of measurements. To avoid the curse of dimensionality, algorithms for feature extraction/selection have been proposed to reduce the amount of data and, at the same time, keep the relevant information necessary to image interpretation or classification [4].

The application of PSO to process hyperspectral data is appealing due to the capability to visualize the location of particles' positions in the search space. Since each spectral dimension corresponds to one band wavelength, the location of the particles' positions may be useful to identify interesting characteristics of the physical process associated with the induction algorithm.

Different approaches for feature selection using PSO have been reported [5], [6]. Nevertheless, the search is commonly limited to a pre-defined number of features, which can be difficult to determine a priori for many problems. In addition, the question of how to define the target functions to be optimized may be highly dependent on the problem at hand.

In this paper, we present a new method for spectral band selection based on PSO. A multi-criteria optimization technique to perform feature selection using two particle swarms is investigated. We developed a method to select optimal spectral bands from hyperspectral data applied on a regression problem in the remote sensing field. Neural networks were implemented to learn models of glucose content in soybeans. Experiments were carried out using real-world hyperspectral datasets from soybean fields.

The authors are with the Interdisciplinary Graduate School of Science and Engineering, Tokyo Institute of Technology, 4259 Nagatsuta, Midori-ku, Yokohama 226-8502, Japan (phone: 81-45-924-5484; fax: 81-45-924-5172; email: monteiro@pms.titech.ac.jp).

II. Hyperspectral Band Selection Algorithm

Feature selection is a subtype of feature extraction where the dimensionality reduction is achieved by selecting bands rather than transforming the data [7]. Feature selection methods are advantageous when the user needs to make decisions based on meaningful features of the original data, or if he wants to exclude non-necessary data components to reduce the cost and labor of data acquisition. Thus, feature selection is highly suitable to hyperspectral imagery, in which the data is intrinsically related to physical wavelengths, and not all spectral bands are always necessary for a certain application.

Assume that the hyperspectral imagery data matrix I is composed of n spectral images $I(\lambda)$, $(\lambda = 1, \ldots, n)$, at each wavelength band λ acquired by the sensor. The aim of feature selection is to find a set of m bands, where $m < n$, to minimize the evaluation criterion.

Feature selection can be implemented as an optimization procedure of search for the optimal feature set that better satisfy a desired measure. We propose a method, as shown in Fig. 1, utilizing two swarms of particles to optimize simultaneously the number of selected features and the error of the model. Each candidate feature set is evaluated by observing its performance on a regression problem. The induction algorithm is a neural network utilized to construct regression models.

A. Particle Swarm Optimization

The PSO algorithm performs optimization in continuous, multidimensional search spaces. PSO starts with a population of random particles, from where the name "particle swarm" is derived. Each particle in PSO is associated with a velocity. Particles' velocities are adjusted according to the historical behavior of each particle and its neighbors while they fly through the search space. Therefore, the particles have a

1-4244-1340-0/07$25.00 ©2007 IEEE

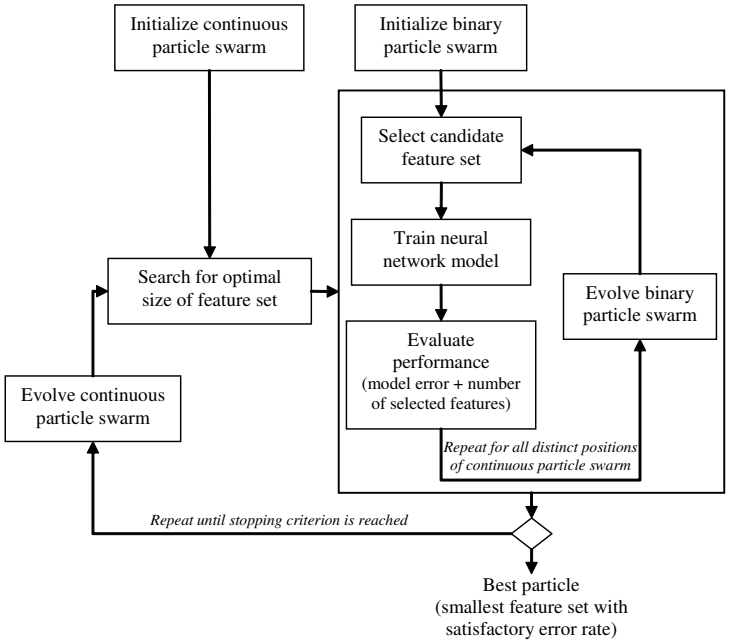


Fig. 1. Diagram of the hyperspectral band selection algorithm based on two particle swarms.

tendency to fly towards the better and better search area over the search process course.

The basic PSO algorithm [8] can be described mathematically by the following equations:

$$v_{id}^{t+1} = wv_{id}^{t} + c_1 r_1^t (p_{id}^t - x_{id}^t) + c_2 r_2^t (p_{gd}^t - x_{gd}^t) \qquad (1)$$

and

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} \ , \qquad (2)$$

where c_1 and c_2 are positive constants, called learning rates; r_1 and r_2 are random functions in the range $[0, 1]$; w is a inertia weight; $X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$ represents the position of the i^{th} particle in a problem space with D dimensions; $V_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$ represents the rate of change of position (velocity); $P_i = (p_{i1}, p_{i2}, \ldots, p_{iD})$ represents the best previous position of the swarm; the index g indicates the best particle among all the particles in the population; and t indicates the iteration number. If the sum of the factors in the right side of Eq. (1) exceeds a specified constant value, particles' velocities on each dimension are clamped to a maximum velocity $V_{\max}$.

Although other approaches for the PSO algorithm have been proposed [9], [10], [11], which may provide faster or even more accurate convergence on specific testbed functions or classes of problems, the PSO version above presents a general competitive performance that is satisfactory for the band selection problem. The first swarm of particles in our method is a "continuous" PSO configured to search for the optimal number of features being selected. The search space of this particle swarm is limited by the number of dimensions of the original dataset. In the case of hyperspectral imagery data, it corresponds to the maximum number of spectral bands available.

B. Binary PSO

To perform the selection of feature sets, the PSO concept needs to be extended in order to deal with binary data. We utilize a binary scheme for feature selection in which each feature is represented by one bit of the particle [12]. If the feature is selected its value is set to 1, if it is not used, it is set to 0.

The candidate feature set is determined using a roulette wheel selection. At each spin of the roulette, the wheel's marker will point to a feature to be selected. The roulette is played until a defined number of selected features is reached. Each feature is assigned with a probability p_{id} proportional to the real value calculated in Eq. (2) limited to the interval [0, 1], according to the equation

$$p_{id} = \frac{x_{id}^{\alpha}}{\sum_{d=1}^{n} x_{id}^{\alpha}} \ , \qquad (3)$$

where α is the selection pressure, which controls the probability of selecting highly fit or less fit features.

The second particle swarm in our method is a "binary" PSO, as described above. Its particles are encoded in n bits, according to the number of dimensions of the dataset.

The feature selection process is carried out in cycles called epochs. In our method, each epoch consists of two phases. Firstly, the continuous particle swarm is evolved, letting the particles update their positions. Then, the second swarm is evolved, each step selecting up to the number of features

defined by the particles of the first swarm. The second swarm may be updated several times at each epoch, for the different positions of the first swarm. However, if two or more particles of the fist swarm are in the same position, only the first occurrence will result in the evolution of the second swarm.

C. Evaluation Function

By simply minimizing the error rate of the induction algorithm, it cannot be expected that the feature selection algorithm will also minimize the number of selected features. We want to search for the smallest feature set that satisfies a desired level of performance of the induction algorithm. For this purpose, the feature selection must be treated as a constrained optimization problem, in which the search is constrained by the size of the feature set and by the specified satisfactory error rate [13].

However, even the binary version of PSO cannot handle this kind of problem directly. We developed a formulation in order to provide control on the balance between the two constraints, necessary when dealing with hyperspectral datasets on regression problems. Otherwise, very small feature sets may be preferred by the algorithm in detriment of possible better performing feature sets with more features.

A performance evaluation function is introduced to accommodate the two constraints, assessing the evolution of the two particle swarms. It can be expressed by the following equation

$$\mathrm{PEF}(x) = k * l(x) + f(e(x)) \ , \tag{4}$$

where x is the candidate feature set selected by the binary particle swarm; l is the cost associated with the size of the feature set, measured by the number of selected features scaled by a constant factor k; and $f(e)$ is a penalty function for the error $e(x)$ of the induction algorithm.

The penalty function defines a region of feasibility of possible solutions in the error space. It can be expressed as

$$f(e(x)) = \frac{\exp((e(x) - u)/s) - 1}{\exp(1) - 1} \ , \tag{5}$$

where u is a feasibility threshold, and s is a small scaling constant.

A feature set is considered feasible if the error in the model output is below the feasibility threshold. For other feature sets presenting higher error, the value of the penalty function grows rapidly.

D. Artificial Neural Networks

We implemented artificial neural networks as the induction algorithm to provide nonlinear models of the regression problem. The nonlinear model was constructed using a multilayer perceptron network, composed of input layer, hidden layer, and output layer, sequentially interconnected in a feed-forward way [14].

The output of the multilayer network can be expressed as $y = f(x) = B\varphi(Ax+a)+b$, where x and y are, respectively, the input and output vectors; A and a are, respectively, the weight matrix and the bias vector of the hidden layer; B and b are, respectively, the weight matrix and the bias vector of the output layer; and φ is the activation function. The activation function for the hidden layer neurons was the hyperbolic tangent sigmoid. The training method was the Levenberg-Marquardt backpropagation [15]. Early stopping was used to improve generalization and avoid overfitting.

The number of neurons in the input layer is proportional to the number of features of the reduced dataset. The neural networks were trained to minimize the mean of squared errors, $\mathrm{MSE}(y) = \frac{1}{N}\sum_{i=1}^{N}(y^o - y^t)^2$, between the test dataset measured values t, and the network outputs o.

III. Results

A. Hyperspectral Dataset

To attest the validity of the proposed method in real-world datasets, experiments were conducted with hyperspectral imagery data from soybean fields. The experimental data was obtained using a hyperspectral sensor, coupled with CCD camera and computer controller. The sensor acquires data in two dimensions, one containing spatial information and, the other, spectral information. In the spatial plane, the hyperspectral camera produces 484 pixels per line. The spectral range comprises the visible to the near-infrared, from 400 nm to 1000 nm, each band interleaved by approximately 5 nm, thus producing 121 spectral bands.

The hyperspectral data was acquired in middle summer on a sunny day, around noontime. The data sample consisted of 13 different varieties of green vegetable soybeans cultivated in an experimental field. In addition, to provide target data for the supervised training of the neural networks, freeze-dried samples from the soybean fields were analyzed in the laboratory using liquid chromatography. The neural networks were trained to model the chemical concentration of glucose in soybeans; the purpose is to predict the sweetness of the soybean crops non-invasively [16].

B. Experiments

The parameters of the particle swarms, shown in Table I, were chosen through experimentation. To define the constants of the penalty function Eq. (5), the error rate of the induction algorithm must be taken into account. The feasibility threshold u must be a value at least slightly greater than the minimum error expected by the best feature set. After preliminary experiments, u was defined as $u = 0.07$. The scaling factor was $s = 5\%$.

The determination of the constant k, in the performance evaluation function Eq. (4), must consider the dimensionality of the problem and the desired performance. If $k = 0$, the PEF value would be equivalent of that of the penalty function alone. When $k = 1$, the PEF value would give a very heavy punishment for acquiring the spectral bands. A more reasonable search space for the hyperspectral dataset problem was obtained by using $k = 0.05$.

The training of the multilayer networks is dependent on the weights' starting values and can be trapped in local

TABLE I

PARAMETERS OF THE PARTICLE SWARMS.

Parameter	Value
Population size continuous swarm	20
Population size binary swarm	40
Learning rate $c_1 = c_2$	2
Maximum particle velocity $V_{\max}$	4
Maximum number of epochs	200
Maximum epochs with constant error	30
Initial inertia w_i	0.9
Final inertia w_f	0.2
Epoch of final inertia	190
Selection pressure α [a]	1

[a]Utilized by the roulette wheel scheme to turn the second swarm into binary.

minima. To minimize this problem, each combination of inputs (candidate feature set) was tested over 3 independent runs and only the best performing networks were retained.

To account for the stochastic nature of the PSO algorithm, the experiments were performed over 10 independent runs for each algorithm, every time initializing the swarms with a different random seed. The evolution of the particle swarms is computationally inexpensive, but the overhead of the feature selection process is in the evaluation of the induction algorithm.

The performance of the particle swarms is presented in Fig. 2. The final particles' positions, assigned to the correspondent spectral band wavelengths, of the 10 runs of the algorithm are shown in Fig. 3. In practice, however, only the best performing feature set selected by the particle swarms is retained, i.e., the feature set presenting the lowest error and highest correlation on the regression problem.

C. Comparison with Principal Components Analysis

Principal components analysis (PCA) is a widely used technique to reduce the dimension of hyperspectral datasets. The PCA algorithm identifies and extracts interesting features by retaining only those components that account for a greater part of the variation in the dataset [17]. The principal components were ordered according to the magnitude of their variance. The variability of the principal components from our soybean dataset is shown in Fig. 4. We set the variance threshold to 99.98%, retaining 11 principal components, same number of features obtained by the particle swarms.

In order to more comprehensibly compare the results between the different methods, the correlation coefficient was calculated as $\mathrm{R}(y) = C(y^o, y^t) \Big/ \sqrt{C(y^o, y^o).C(y^t, y^t)}$, where C is the covariance matrix, and o and t indicate the neural network output and the test dataset measurement, respectively. A summary of the results comparing the proposed method, the best feature set selected by the particle swarms, and the PCA is presented in Table II.

TABLE II

COMPARISON OF PARTICLE SWARMS FEATURE SELECTION (PSO-FS) AND PCA, APPLIED TO MODEL GLUCOSE CONTENT IN SOYBEAN CROPS FROM HYPERSPECTRAL DATA USING NEURAL NETWORKS.

Algorithm	PEF[a]	MSE[b]	R[c]
PSO-FS	0.6382	0.0130	0.8620
PCA		0.0149	0.8425

[a]Performance evaluation function
[b]Mean squared error
[c]Pearson's correlation coefficient

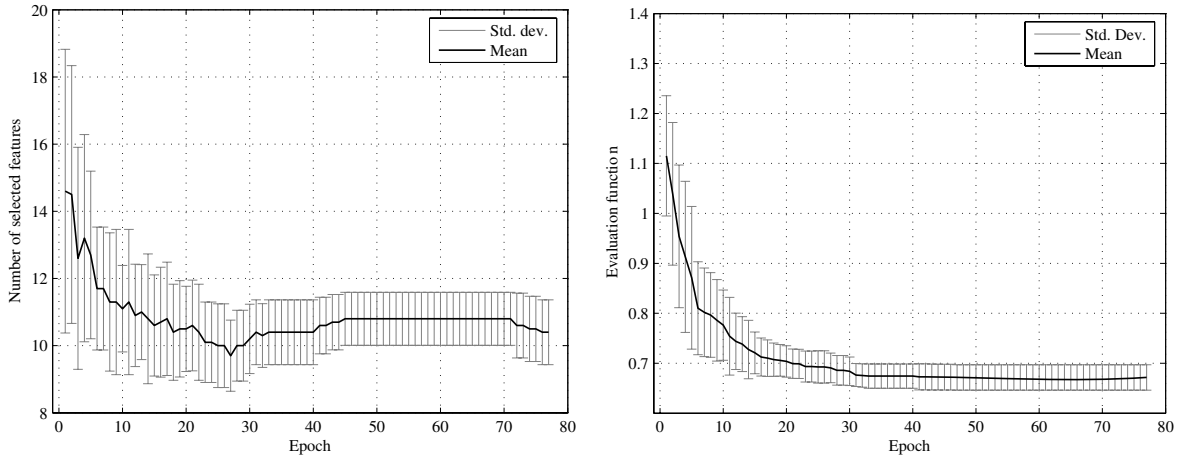
IV. CONCLUSIONS

This paper proposes a feature selection method based on two particle swarms, a continuous and a binary, to search not only for the optimal feature set, but also for the optimal number of features, at the same time. Furthermore, the applicability of the method to extract information from hyperspectral imagery data was demonstrated. The method was successfully validated with experiments utilizing real-world datasets of soybean fields applied on a regression problem. The particle swarms were implemented in conjunction with neural networks to model the sweetness in soybean crops, a non-trivial problem.

The particle swarms were able to optimize the combined criteria efficiently. In spite of the limited size of the particle swarms' populations, the proposed algorithm was capable of fast convergence towards the optimal region of the search space. The particle swarms outperformed the PCA in our experiments. However, despite the deceiving impression that just 4 principal components hold most of the variability in the soybeans dataset, the rather close results presented by PCA were only achieved using a higher number of principal components, 11.

We developed a performance evaluation function adapting the PSO algorithm to search for the optimal feature set while constrained by two criteria, the error rate of the induction algorithm and the size of the feature set. The performance evaluation function punishes feature sets with high dimensionality. This function may produce excessive punishment, particularly on real-world hyperspectral imagery data, causing the selection of small feature sets presenting undue error. Thus, in order to determine a better compromise between the number of selected features and the induction algorithm's error rate, a constant factor k in Eq. (4) was introduced in this paper.

The particle swarms also possess the advantage of permitting the visualization of the selected features in contrast with their spectral locations, providing an appealing analysis tool for the field of remote sensing. The selected band wavelengths tended to be spatially distributed over the spectra, resulting in an efficient use of the available information. However, some spectral regions are consistently preferred by most of the particle swarms, specifically in the range between 500 nm and 820 nm. We propose the method not only for



(a) Number of selected features optimized by the first swarm

(b) Performance evaluation function minimized by both swarms

Fig. 2. Curves of the evolution of the particle swarms showing mean value and standard deviation over 10 runs.

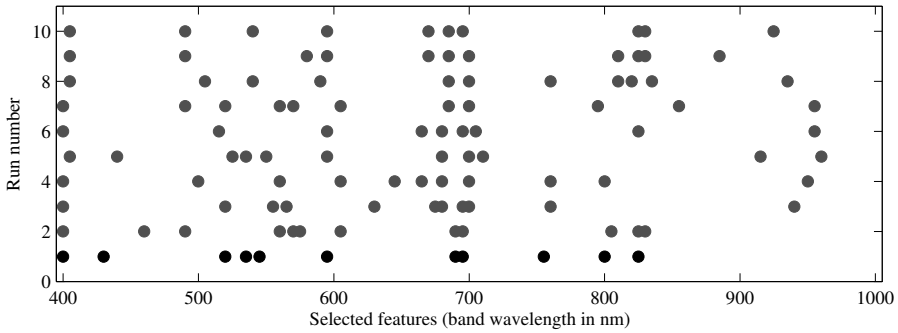


Fig. 3. Spectral location of the feature sets selected by the particle swarms over 10 runs. The best performing feature set is indicated in dark black.

dimensionality reduction, but also as a valuable tool for the spectral analysis of remotely sensed hyperspectral imagery.

Future works will involve improving the performance and accuracy of the proposed method. Different architectures for the PSO algorithm could be evaluated to identify which version may produce better results on the band selection problem. Since the fitness evaluation based on the neural network is the main time consuming component of the search process, a more efficient training method could be implemented, perhaps using a fast PSO approach.

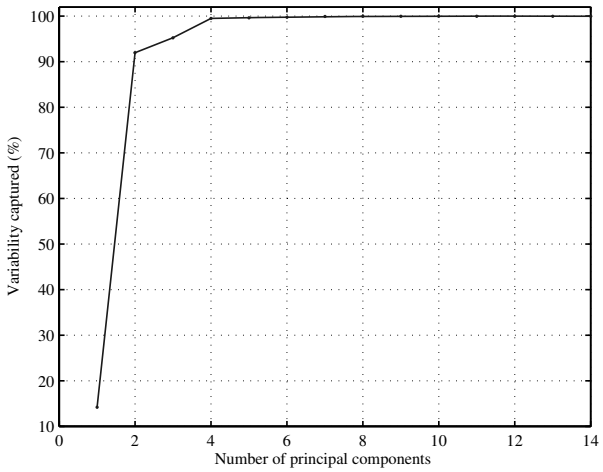


Fig. 4. PCA variability for the hyperspectral dataset of soybeans

ACKNOWLEDGMENT

We would like to thank Mr. Yohei Minekawa of the Tokyo Institute of Technology for his help with the experimental data. We also thank Dr. Keisuke Kameyama of the University of Tsukuba for the discussions about the PSO algorithm. This research is part of a joint collaboration with Prof. Tsuneya Akazawa of Yamagata University and Mr. Kunio Oda of the Yamagata General Agricultural Research Center.

REFERENCES

[1] J. Kennedy and R. C. Eberhart, *Swarm Intelligence*. San Francisco: Morgan Kaufmann Publishers, 2001.

[2] P. M. Mather, *Computer Processing of Remotely-Sensed Images, An Introduction*. Chichester: John Wiley & Sons, 2004.

[3] R. E. Bellman, *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 1961.

[4] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research, Special Issue on Variable and Feature Selection*, vol. 3, pp. 1157–1182, 2003.

[5] H. A. Firpi and E. Goodman, "Swarmed feature selection," in *Proc. 33rd Applied Imagery Pattern Recognition Workshop*, 2004, pp. 112–118.

[6] Y. Liu, Z. Qin, Z. Xu, and X. He, "Feature selection with particle swarms," in *Proc. Computational and Information Science*, vol. 3314, 2004, pp. 425–430.

[7] H. Liu and H. Motoda, "Feature transformation and subset selection," *IEEE Intelligent Systems, Special Issue on Feature Transformation and Subset Selection*, vol. 13, no. 2, pp. 26–28, 1998.

[8] Y. Shi and R. C. Eberhart, "A modified particle swarm optimizer," in *Proc. IEEE Congress on Evolutionary Computation*, 1998, pp. 69–73.

[9] M. Clerc and J. Kennedy, "The particle swarm - explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2002.

[10] F. van den Bergh and A. P. Engelbrecht, "A cooperative approach to particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 225– 239, 2004.

[11] R. Mendes, J. Kennedy, and J. Neves, "The fully informed particle swarm: simpler, maybe better," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 204– 210, 2004.

[12] D. K. Agrafiotis and W. Cedeño, "Feature selection for structure-activity correlation using binary particle swarms," *Journal of Medicinal Chemistry*, vol. 45, pp. 1098–1107, 2002.

[13] W. Siedlecki and J. Sklansky, "A note on genetic algorithms for large-scale feature selection," *Pattern Recognition Letters*, vol. 10, no. 5, pp. 335–347, 1989.

[14] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed. Englewood Cliffs: Prentice-Hall, 1999.

[15] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the marquardt algorithm," *IEEE Transactions on Neural Networks*, vol. 5, no. 6, pp. 989–993, 1994.

[16] S. T. Monteiro, Y. Minekawa, Y. Kosugi, T. Akazawa, and K. Oda, "Prediction of sweetness and nitrogen content in soybean crops from high resolution hyperspectral imagery," in *Proc. 2006 IEEE International Geoscience and Remote Sensing Symposium*, vol. 5, Denver, Colorado, 2006, pp. 2263–2266.

[17] I. T. Jolliffe, *Principal Component Analysis*. New York: Springer-Verlag, 1988.