# 行政院國家科學委員會專題研究計畫成果報告

## 超光谱影像之快速鑑別技術 （一）

## Real-Time Hyperspectral Imaging Identification (I)

Spectrogram Identification By Gaussian Deconvolution and Neural Learning

計畫編號：NSC 89-2212-E-002-038

執行期限：88 年 8 月 1 日至 89 年 7 月 31 日

主持人：林啟萬 國立台灣大學電機工程學系

計劃參與人員：吳震乙 蔡仁彥 謝楊坤 國立台灣大學電機工程學系

**中文摘要**

利用超多頻遙感影像鑑別特定區域的構成物質，在地質、農業和軍事方面有很大的用途，在技術上，鑑別物質的能力必須由辨認或鑑別所觀測到的光谱來實現，針對這種應用，本研究利用光谱圖內所隱涵的吸收帶波長和強度做為類神經網的輸入，成功的發展出一個類神經網光谱圖鑑別器，萃取吸收帶波長和強度的方法是結合濾波微分分析、限制型高斯近似以及高斯近似的連續過程，鑑別功能則用庫倫能類神經網的學習和回憶能力來實現，用此光谱圖鑑別器鑑別噴墨印製的各色圖像和五種不同物品的光谱，其實驗結果甚為準確，應可符合實用的需求。

**關鍵詞：超多頻、光谱、遙感、鑑別、高斯近似**

**Abstract**

To characterize the surface of a target site for geological, military or agricultural applications by remote hyperspectral imaging requires identifying or classifying the observed spectra. By extracting the band wavelength positions and strengths from the interested spectrograms as inputs to a neural network, this research successfully develops a neural spectrogram identifier. The extraction of absorption features is a successive procedure of filtered derivative analysis, constrained Gaussian function approximation and Gaussian function approximation. Identification is implemented by using a Restricted Coulomb Energy neural network to learn the interested spectrograms by discriminating their absorption features and then retrieves the categories representing the spectrograms. Experimentally identifying different colors of an ink-jet printed sample image and the hyperspectral data of five categories of objects have validated the whole design.

**Keywords**：hyperspectral imaging, spectrum, classification, identification, Gaussian approximation, remote sensing

## 1. Introduction

Hyperspectral imaging systems will provide the opportunity for remote identification of a target site. The hyperspectral imaging with high spatial resolutions will facilitate detailed surface characterization. Spectral resolution will affect the quality and quantity of geological information derivable from optical remote sensing imagery. The higher spectral resolution data provides in general greater classification accuracy [Hunt 1978, Blom 1980, Kruse 1989, Ben-Dor 1994]. However a hyperspectral

imaging system usually acquires hundreds of bands of spectral information for each pixel in a scene. For the tremendous volumes of resultant data, real-time identifying the underlying spectral signatures in scene with on-board processor makes possible the collection of a larger data set, and the downlink useful products to the ground in real-time operation [Curtiss 1997]. The quantitative information for identification can be extracted from spectral parameters such as band wavelength positions, strengths, widths, areas and absolute reflectance values. Conventional methods are derivative analysis, red edge derivative analysis, spectral matching, end-member mixing models, maximum likelihood classification, principal components and canonical analysis, linear regression, ratio analysis, spectral decomposition, spectral de-convolution, theoretical treatments, and modified Gussian model analysis [Cloutis 1996] [Staenz 1992] [Palmadesso 1995]. Identification or classification criteria can be the spectral absorption bands, the presence of specific spectral features, match of the coefficient of polynomial type expansion of the spectral signature as a function of wavelength over a wide spectral range. Identification requires a detailed knowledge of the spectral signatures of the materials being sought, as well as of all the other materials in the scene, and the development of a spectral signature library of all expected natural materials [Elachi 1987]. Here inspired by the extensively use of estimating the absorption features by de-convoluting the composite spectrum into individual Gaussian or modified Gaussian distributions [Clark 1984] [Huguenin 1986] [Sunshine 1990, 1993] [Crilly 1991] [Miekina 1997], we propose to classify hyperspectral images by identifying the spectrograms of interested materials. The spectral reflectance signatures are assumed being converted and recorded as absorption spectrograms. Gaussian model analysis is applied to extract the absorption features from the interested spectrograms. Then a Restricted Coulomb Energy (RCE) neural network learns to identify each spectrogram by discriminating their absorption features and thereafter the trained neural network can find out similar spectrograms appearing in the interested hyperspectral images. The applications can be identification if the corresponding materials of the interested spectrograms are known or simply classification if no knowledge of the materials.

## 2. Problem formulation

The considered problem is that given a uniformly error-corrupted spectrogram, modeled with real-valued sequence:

$$y_k = y_{ideal}(x_k) + \sup\left\{\left|y_{ideal}(x)\right|\right\}\varepsilon_k \quad \text{for } k = 1, 2, \cdots, m \qquad (1)$$

where $x_k \in [x_{min}, x_{max}]$ and $\varepsilon_k$ are realizations of identical zero-mean random variables. First perform Gaussian model analysis to parameterize $y_{ideal}(x)$ as

$$y_{ideal}(x) \approx g(x, w) = \sum_{i=1}^{n}\left\{s_i \exp\left(\frac{-(x-\mu_i)^2}{2\sigma_i^2}\right)\right\} \qquad (2)$$

where $n$ is the number of absorption bands or Gaussian distributions, $\mu_i$ the band wavelength positions or Gaussian peaks, $\sigma_i$ the standard deviation, $s_i$ the strength, and $w$ the parameter vector as

$$w = [\mu_1, \sigma_1, s_1, \mu_2, \sigma_2, s_2, \cdots, \mu_n, \sigma_n, s_n]^T \qquad (3)$$

Then design a neural network and its training procedure to identify the interested spectrograms in the hyperspectral images by learning their absorption features.

## 3. A multi-step algorithm of Gaussian approximation

Represent the digitized spectrogram as

$$Y = [y_1, y_2, \cdots, y_m]^T$$

and its approximation as

$$G = [g(x_1, w), g(x_2, w), \cdots, g(x_m, w)]^T$$

The criterion of Gaussian approximation is to solve for the parameter vector $w$ by minimizing the following cost function

$$\phi(w) = \|(Y - G)\|_2^2 \qquad (4)$$

Since Gaussian distribution functions are nonlinear and bi-orthogonal; the minimization is therefore a procedure of nonlinear procedure. Conventionally the steepest deacent method may be used to find a local minimum of (4) and it will usually converge even for poor initial approximations, except the speed of convergence is linear [Burden 1997]. As an improvement, Levenberg-Marquardt (LM) method was developed to find a local minimum more efficiently by shifting between the steepest descent and the Hessian iterative methods [Sunshine 1990]. However without sufficient accurate starting values neither the steepest descent nor LM method will assure to converge to the global minimum. To achieve the global minimum, a multi-step algorithm of Gaussian approximation is proposed as depicted in Fig. 1. Firstly derivative analysis is applied to detect the band wavelength positions. Secondly, using these band positions as Gaussian peaks, the constrained Gaussian approximation estimates the gross strengths and widths. Finally starting with the gross Gaussian parameters, the Gaussian approximation fine adjusts all the parameters. This multi-step algorithm enables the constrained Gaussian and the follow up Gaussian approximations to start with appropriate values so that the global minimum will be achieved.

### 3.1 Detecting the band wavelength positions

Derivative analysis has been used in spectral de-convolution to detect the wavelength positions of the absorption bands [Sunshine 1990]. Using the results of derivative analysis on the absorption spectrogram as constraints to the Gaussian approximation can obtain Gaussian parameters equivalent to the absorption features. The band wavelength positions in an absorption spectrogram can be detected at [Huguenin 1986] [Wu 1999]

$$\overline{\mu}_i = x \text{ for all } y^{V}(x) = 0,\ y^{II}(x) < 0, \text{and } y^{IV}(x) > 0 \qquad (5)$$

where $x \in [x_{min}, x_{max}]$, $\overline{\mu}_i$ is the estimated position, and the superscript denotes the order of derivative. To calculate $y^{V}(x)$, $y^{II}(x)$ and $y^{IV}(x)$ efficiently on the error-corrupted data, techniques of smoothing and differentiation such as polynomial curve fitting [Huguenin 1986], mean filter smoothing [Kawata 1984], finite approximation [Tsai 1997], and simultaneous smoothing and differentiation [Savitzky 1964][King 1999] were developed. In general when the absorption bands are broad and the noise is relatively high frequency, mean filter smoothing is a better choice than least-square fit smoothing. Finite approximation is better than polynomial curve fitting in that computation is straightforward and less possible

to introduce artifacts. This research develops two filters to approximate the first and second order differentiations. The filters are simultaneous procedures of mean filter smoothing and finite divided difference. The mean filter simply takes the average of all points in a moving window as the new value of the middle point as follows:

$$\hat{y}(x_k) = \frac{\sum_{i=-n}^{n} y(x_{k+i})}{2n+1} \quad (6)$$

where the window size is $2n+1$ points and '$\hat{y}$' denotes the filtered value. Finite divided difference approximates the first order derivative as

$$\left.\frac{dy}{dx}\right|_{x_k} \approx \frac{\hat{y}(x_{k+d}) - \hat{y}(x_{k-d})}{x_{k+d} - x_{k-d}} = \frac{\Delta y}{\Delta x} \quad (7)$$

where $x_{k+d},\ldots,x_{k-d}$ are points within the separation window. The convolution of the mean filter and the finite divided difference functions obtains a filter, denoted as $f_1$, that can approximate the first order differentiation as

$$f_1(x_i) = \frac{\{u[k-(-d+n+1)]-u[k-(d+n+1)]\}-\{u[k-(-d-n)]-u[k-(d-n)]\}}{(2n+1)(x_{i+d}-x_{i-d})},\ n \geq d \quad (8)$$

where $u(\bullet)$ denotes the input. The approximated first order derivative is obtained as

$$\hat{y}^I(x_i) = \sum_{i=-n-d}^{n+d} y(x_{k-i}) f_1(x_i) \text{ for } (n+d) < k < (m-n-d+1) \quad (9)$$

$$\hat{y}^I(x_k) = \hat{y}^I(x_{n+1+d}) \text{ for } 1 \leq k \leq (n+d) \quad (10)$$

and

$$\hat{y}^I(x_k) = \hat{y}^I(x_{L-n-d}) \text{ for } (m-n-d+1) \leq k \leq m \quad (11)$$

The filter, $f_2$, to approximate the second order differentiation is obtained by taking the convolution of two $f_1$ filters as

$$f_2(x_i) = \frac{[2d-|k+2n+1|]\cdot\{u[k+(2n+2d)]-u[k+(2n-2d+1)]\}}{\{(2n+1)(x_{i+d}-x_{i-d})\}^2}$$
$$+\frac{[(-4d)+|2k|]\cdot\{u[k-(-2d)]-u[k-(2d)]\}}{\{(2n+1)(x_{i+d}-x_{i-d})\}^2}$$
$$+\frac{[2d-|-k+2n+1|]\cdot\{u[k-(2n-2d+1)]-u[k-(2n+2d)]\}}{\{(2n+1)(x_{i+d}-x_{i-d})\}^2} \quad (12)$$

The approximated second order derivative is obtained as

$$\hat{y}^{II}(x_k) = \sum_{i=-2n-2d}^{2n+2d} y(x_{k-i}) f_2(x_i) \text{ for } (2n+2d) < k < (m-2n-2d+1) \quad (13)$$

$$\hat{y}^{II}(x_k) = \hat{y}^{II}(x_{2n+1+2d}) \text{ for } 1 \leq k \leq 2n+2d \quad (14)$$

and

$$\hat{y}^{II}(x_k) = \hat{y}^{II}(x_{L-2n-2d}) \text{ for } m-2n-2d+1 \leq k \leq m \quad (15)$$

To compute $y^{II}(x)$, $y^{IV}(x)$ and $y^{V}(x)$ for detecting the band wavelength positions, one can cascade $f_2$, $f_2$ and then $f_1$ to process the error-corrupted data. The computations will require performing five convolutions instead of ten for that without using simultaneous smoothing and differentiation. The derivative analysis estimates the band wavelength positions. Represent the results as

$$w^{(0)} = [\bar{\mu}_1, \sigma_1^{(0)}, s_1^{(0)}, \bar{\mu}_2, \sigma_2^{(0)}, s_2^{(0)}, \cdots, \bar{\mu}_n, \sigma_n^{(0)}, s_n^{(0)}]^T \quad (16)$$

where $n$ and $\bar{\mu}_i, i = 1, 2, \cdots, n$ are values obtained in this step, and the superscript '(0)' denotes the items to be determined later.

Example 1: Researchers frequently used Huguenin's test data as a benchmark to test extraction of absorption features [Huguenin 1986]. The data set are generated by using the following equations:

$$x = 5000 + k/1024 * 20000 \text{ for } k = 1,2,3,\ldots,1024 \quad (17)$$

$$y(x_k) = g(x_k) = \sum_{i=1}^{6} g_i(x_k) \quad (18)$$

Six Gaussian distribution functions (absorption bands) are contained in the simulated spectrogram, and the parameters are listed in Table 1. Figure 2 shows the spectrogram

(absorbance) of the Huguenin's test data corrupted with white noise of SNR=25dB (signal-to-noise-ratio). For simultaneous smoothing and differentiation, trade-off is required between removing more noise and keeping the signal detail. Simulations have been conducted for ($d$, $n$)= (10, 10), (10, 15), (10, 20) and (10, 25), respectively. For the first two cases, the values of derivatives are far from correct, and for the last one the detail of the spectrogram may be lost. The one acceptable was obtained for ($d$, $n$)= (10, 20). Fig. 3 shows the impulse responses of the corresponding $f_1$ and $f_2$. The three derivatives are presented in Fig. 4 and the band wavelength positions (Gaussian peaks) are marked at the six zero-crossings of the fifth derivative. Table 1 also compares the estimated positions of the six Gaussian peaks with the correct values.

3.2 The constrained Gaussian and Gaussian approximations

Theoretically the Gaussian parameters, $w$, can be estimated by minimizing the cost function (4). But since Gaussian distribution function is nonlinear, iterative optimization by satisfying

$$\nabla\phi(w) = -2J_{G^T}(Y - G) = 0 \qquad (19)$$

where '$\nabla$' denotes taking gradient and $J_{G^T}$ is the Jacobian matrix of the vector $G^T$, may be diverge or trapped in a local minimum. To achieve the convergence of global minimum, the optimization applies first a constrained Gaussian approximation and then followed by a Gaussian approximation. The constrained Gaussian approximation accepts the band wavelength positions obtained by derivative analysis as the Gaussian peaks and keeps them unchanged. The strengths and widths that are initially set arbitrarily are the only parameters to be adjusted. The starting parameter vector is represented as (16), where $n$ and $\overline{\mu}_i, i = 1 \text{ to } n$ is kept unchanged, the superscript '(0)' denotes iteration and $\sigma_i^{(0)}$ and $s_i^{(0)}$, $i = 1 \text{ to } n$ are arbitrary nonzero values. In the constrained Gaussian approximation the iterative rule should be of good convergence for poor starting values. For this purpose the steepest descent method is a good candidate. Subject to the constraint on the Gaussian peaks and assuming the descent rate of the steepest descent rule is small enough, the convergent result denoted as

$$\overline{w} = [\overline{\mu}_1, \overline{\sigma}_1, \overline{s}_1, \overline{\mu}_2, \overline{\sigma}_2, \overline{s}_2, \cdots, \overline{\mu}_n, \overline{\sigma}_n, \overline{s}_n]^T \qquad (20)$$

is near the global minimum. Using $\overline{w}$ as the starting values of the follow up Gaussian approximation, iterative optimization such as Hessian's method or LM method can be applied to fine adjust the whole parameter vector.

3.3 Iterative rules of nonlinear optimization

Conventionally the steepest descent method of nonlinear optimization obtains iterative rule as

$$w^{(J+1)} = w^{(J)} - \delta^{(J)} \nabla\phi(w^{(J)}) \qquad (21)$$

where the parenthesized superscript denotes the iterations, and $\delta^{(J)}$ is a fixed or adaptable descent rate to control the convergence and rate of convergence. The steepest descent rule will usually converge even for poor starting values, except the speed of convergence is linear and a slight difficulty arises from no knowledge of choosing the descent rate to achieve fast descent and accurate result.

When starting values are sufficiently near the global minimum, minimization of the cost

function (4) can be transformed into a root-finding problem for

$$F = \frac{-1}{2}\nabla\phi(w) = J_{G^T}(Y - G) = 0 \qquad (22)$$

In this case Newton's method that gives quadratic convergence obtains an iterative formula as follows [Burden 1997]:

$$w^{(J+1)} = w^{(J)} - J^{-1}_{F^{(J)}} F^{(J)} = w^{(J)} - J^{-1}_{\nabla\phi(w^{(J)})} \nabla\phi(w^{(J)}) \qquad (23)$$

Although there are methods to reduce the computation burden of (23), to solve for the Jacobian matrix and its inverse is in general complicated. A simplified approach is using the Hessian iterative method by assuming

$$F^{(J+1)} \approx J^{(J)}_{G^T}\left(Y - G^{(J+1)}\right) = 0 \qquad (24)$$

and expanding $G^{(J+1)}$ in a Taylor series and ignoring the terms higher than first-order as

$$G^{(J+1)} \approx G^{(J)} + J^{(J)}_{G}\left(w^{(J+1)} - w^{(J)}\right) \qquad (25)$$

Substituting (25) into (24) and after rearrangement gives the Hessian iterative rule as follows:

$$w^{(J+1)} = w^{(J)} - \frac{1}{2}\left\{J^{(J)}_{G^T} J^{(J)}_{G}\right\}^{-1} \nabla\phi(w^{(J)}) \qquad (26)$$

With starting values sufficiently near the global minimum, (26) can obtain quadratic convergence of the Gaussian approximation. However there will usually be no guarantee to the starting values, therefore LM method gets popular applications in nonlinear optimization [Sunshine 1990]. LM method can be formulated as follows:

$$w^{(J+1)} = w^{(J)} - \frac{1}{2} D^{(J)^{-1}} \nabla\phi(w^{(J)}) \qquad (27)$$

where

$$D^{(J)} = J^{(J)}_{G^T} J^{(J)}_{G} + \lambda I \qquad (28)$$

and where $I$ denotes the unitary matrix. LM method has a procedure to adapt the value of $\lambda$ between zero and infinity. When $\lambda$ is large, $D^{(J)}$ will be dominated by $\lambda I$, and (27) becomes almost the steepest decent rule. Conversely when $\lambda$ is small, (27) will approximate the Hessian iterative rule (26). But practically in the iterative procedure of LM method, $\lambda$ can certainly be small but never go to zero. This small $\lambda$ makes it slightly different from the Hessian iterative rule and may introduce small error into the final result of optimization. If one tries to apply LM method to the constrained Gaussian approximation starting with arbitrary setting to the strengths and widths, there will be no guarantee to the result. Most frequently the procedure may be stopped due to singularity of matrix inversion or sometimes trapped in a local minimum.

Example 2: Using the results of derivative analysis in Table 1 of Example 1 to construct $w^{(0)}$ as shown in Table 2 where $\omega_i^{(0)}$ and $s_i^{(0)}$ $i$=1 to 6 are arbitrary settings. With $w^{(0)}$ the cost function gives $\phi(w^{(0)}) = 1.7719 \times 10^6$. Using the steepest descent rule (21) with constant descent rate, $\delta^{(J)} = 0.05$, and $\phi(w^{(J+1)}) \geq \phi(w^{(J)})$ as stopping criterion for the constrained Gaussian approximation, the procedure had run for 52 iterations. The results are shown as $\overline{w}$ in Table 2 and at this time the value of the cost function is $\phi(w^{(52)}) = 3.4480 \times 10^5$. Finally starting with $\overline{w}$ the Hessian iterative rule is applied to solve for the Gaussian approximation. The results are shown as $\hat{w}$ in Table 2 and the cost function reads $\phi(w^{(59)}) = 7.8887 \times 10^{-7}$. At the end of the multi-step approximation small errors

appear only in the FWHM's and the algorithm is surely successful. It should be mentioned that starting with $w^{(0)}$ and no matter using or skipping the step of constrained Gaussian approximation, LM method did not achieve an acceptable result.

## 4. Neural identification of spectrograms

For the identification of an object material, it will be assumed that the organization of the band wavelength positions and strengths of its absorption spectrogram in some waveband is unique. Then a RCE neural network is developed to identify the interested spectrogram by using these band wavelength positions and strengths as input data (i.e. width is not used).

### 4.1 The architecture of the RCE neural network

RCE neural networks learn to map an arbitrary input feature vector from real-valued $n$-dimensional space to an output category space. A basic RCE network is illustrated in Fig. 5. The network is a reduced connection, feedforward neural network with three processing layers: an input layer with $n$ nodes, a hidden layer consisting of $m$ nodes, and an output layer with $c$ nodes [Reily 1982, 1987]. Nodes in the input layer correspond to feature or attribute values of the input pattern, whereas nodes in the output layer each correspond to a different distinct class or category. Each hidden-layer node is fully connected to all input-layer nodes through fixed weights and connects to only a single output-layer node. There is no lateral or feedback connection. Nodes in the hidden layer map the input feature values to the output categories. Input patterns that activate a single output response are called well-decided mappings. Input vectors that activate multiple output responses or no output response are ambiguous mappings. After training, the network should give well-decided outputs.

Each hidden layer node defines a region of "influence" in pattern space based on its input weight vector $v$ and an associated threshold parameter $\theta$. The region is defined through an activation function

$$F(d)=\begin{cases}F(d) & \text{if } d<\theta \\ 0 & \text{if } d\geq\theta\end{cases} \qquad (29)$$

where $d=d(v,x)$ is an appropriately chosen metric. The metric can be Cartesian distance, Hamming distance, vector inner product or some other measure, and $x$ is the input feature vector. Let $d$ be the Cartesian distance in $n$-dimensional real-valued space, the region of influence will be a hypersphere of radius $\theta$ with center located at $v$. To summarize, the region of influence for a given mapping layer node $i$ is defined by (1) a central point in $n$-space determined by the node's weihgt vector $v$, (2) the size (radius) of the region determined by the threshold parameter $\theta$ and (3) the topology or geometry of the region determined by the metric $d$.

Each hidden node in a well-trained RCE network represents one prototype and is called a prototype node. Each prototype node can have fuzzy property by defining its fuzzy sets together with their corresponding membership functions [Roan 1993]. The triangular membership functions defined below are easy for applications,

$$\eta(x)=\begin{cases}(1/b)(-|x-a|+b) & \text{if } (a-b)<x<(a+b) \\ 0 & \text{otherwise}\end{cases} \qquad (30)$$

where $a$ is the center and $b$ is the width of the

triangular membership function. If the input dimension is $n$, then $n$ fuzzy membership functions $\{ \eta_{ij}, j = 1,2,3.....n \}$ should be defined for prototype node $i$ (denoted by $p_i$). For simplicity, only one common width $b_i$ is used in prototype node $p_i$. Then each prototype node $p_i$ has $(n+1)$ parameters: $a_{i1}, a_{i2}, .....a_{in}$, and $b_i$ as shown in Fig. 6.

4.2 Algorithm of training the RCE neural network

For each training input vector $x = (x_1, x_2, .....x_n)$ belonging to the $k^{th}$ category, perform the following steps.

Step 1: Calculate the output of all prototype nodes $p_i$ by

$$OUT_{p_i} = \Phi[\frac{1}{n}\sum_{j=1}^{n}\eta_{ij}(x_j)] \quad (31)$$

where $\eta_{ij}$ is defined by (30) and

$$\Phi(z) = \begin{cases} z & \text{if } z > \theta \\ 0 & \text{if } z \le \theta \end{cases} \quad (32)$$

where $\theta$ is a prespecified threshold. If $OUT_{p_i} > 0$, then prototype node $p_i$ is activated.

Step 2: If none of the prototype nodes in the $k^{th}$ category (i.e., the prototype nodes connected to the $k^{th}$ output node) is activated, then create a new prototype node $p_i$ which connects to the $k^{th}$ output node and set $a_i = (a_{i1}, a_{i2}, ..., a_{in}) = x, b_i = b_0$, where $b_0$ is a predefined initial value for all membership functions.

Step 3: If any prototype node $p_i$ that does not connect to the $k^{th}$ output node is activated, then reduce $b_i$ to $b_i'$ such that $p_i$ cannot be activated. The value of $b_i'$ is calculated by

$$b_i' = \frac{\sum_{j=1}^{n} f(x_i)}{(n-k) - n\theta} \quad (33)$$

where

$$f(x_j) = \begin{cases} |x_j - a_{ij}| & if\ (a_{ij} - b_i) < x_j < (a_{ij} + b_i) \\ 0 & otherwise \end{cases} \quad (34)$$

and $k$ is the number of zero of $f(x_j)$.

In retrieving, we first calculate the output of all prototype nodes using (31) and (32) and then find the prototype node with the maximum output value. If this node is connected to the $k^{th}$ output node, the testing input vector is assigned to category $k$. If no prototype node is found, then the input vector is undefined.

The RCE neural networks have important features such as self-growing to appropriate size, no local minima problem, low computation requirements, high storage efficiency, low connectivity, and the ability to learn quickly. However they also possesses poor qualities such as poor in generalization, growing very large for some problems and they cannot perform general mapping function tasks.

4.3 A neural spectrogram identifier

Using a RCE neural network as the core, the following paragraphs develop a neural identifier of absorption spectrograms that accepts the extracted Gaussian parameters as inputs.

4.3.1 Input to the neural spectrogram identifier

Only the Gaussian peaks and strengths, simply mentioned as Gaussian pairs, are chosen as the input, FWHM is not included. Considering the Gaussian Deconvolution with derivative analysis is noise sensitive and may result in some faulty minor Gaussian

distributions. Only the significant Gaussian distributions (i.e. the significant absorption bands) are selected for the spectrogram identification. Zeros may be filled in to make all the inputs have the same number of Gaussian pairs. Regarding the Gaussian peaks and strengths of the interested spectrograms as two feature sets, denoted as $f_\mu$ and $f_s$, respectively, and each of them has $M$ elements. Then an input pattern, $x$, to the neural spectrogram identifier is represented as the following row vector

$$x = [x_1, x_2, \ldots x_M, x_{M+1}, \ldots, x_{2M}] = [\mu_1, \mu_2, \ldots \mu_M, s_1, s_2, \ldots s_M] \quad (35)$$

where $\mu_j$ denotes the $j^{th}$ peak and $s_j$ the $j^{th}$ strength. Since the Gaussian peaks and strengths are mostly distributed in different scales, the widths of the fuzzy membership functions for them should be defined individually. Then each prototype node has $2 \times M + 2$ parameters for the membership functions as

$$\mathbf{a} = [a_1, a_2, \ldots, a_{2M}], \; b_\mu, \; b_s \quad (36)$$

where $b_\mu$ and $b_s$ are used to remember the widths of the membership functions for the peak and strength, respectively.

In the procedure of identification, each input Gaussian pair is compared with that of the prototype. The resulted output is the similarity, $\varepsilon_j$, of the input to the prototype as

$$\varepsilon_j = v_\mu \times \eta_j(\mu_j) + v_s \times \eta_{j+M}(s_j) \quad (37)$$

where $v_\mu$ and $v_s$ are weights to the similarity of the peak and the strength, respectively, and $\eta_j$ is the $j^{th}$ triangular membership function defined as

$$\eta_j(x) = (1/b_\mu)(-|x - a_j| + b_\mu) \quad \text{if } x \in f_\mu \text{ and } (a_j - b_\mu) < x < (a_j + b_\mu)$$

$$\eta_j(x) = (1/b_s)(-|x - a_j| + b_s) \quad \text{if } x \in f_s \text{ and } (a_j - b_s) < x < (a_j + b_s)$$

and

$$\eta_j(x) = 0, \text{ otherwise} \quad (38)$$

The output of a prototype node is calculated as the average of the similarities to the input spectrogram as follows:

$$OUT_p = \Phi[\frac{1}{M}\sum_{j=1}^{M} \varepsilon_j] \quad (39)$$

where $\Phi$ is defined as (32).

4.3.2 Training and retrieving the neural spectrogram identifier

For an input of the $k^{th}$ category represented as

$$x_0 = [\mu_1, \mu_2, \ldots \mu_{M_x}, 0, \ldots 0_{M_{MAX}}, s_1, s_2, \ldots s_{M_x}, 0, \ldots 0_{M_{MAX}}] = [x_1, x_2, \ldots x_{2M_{MAX}}]$$

Define the following parameters appropriately:

$\theta$ : the threshold which decides whether a prototype will be activated or not,

$b_{s0}$: the initial width of the membership function for the Gaussian strength,

$b_{\mu 0}$: the initial width of the membership function for the Gaussian peak,

$\Delta b_s$: the step size to reduce $b_{si}$ when prototype $p_i$ is activated incorrectly,

$\Delta b_\mu$: the step size to reduce $b_{\mu i}$ when prototype $p_i$ is activated incorrectly.

$v_\mu$ : the weight to the similarity of the

Gaussian peak.

$v_s$ : the weight to the similarity of the Gaussian strength.

Then the training is summarized as the following steps:

Step 1: Calculate the outputs of all prototype nodes, $p_i$, by

$$\varepsilon_{ij} = w_\mu \times \eta_{ij}(\mu_j) + w_s \times \eta_{i,j+M_{MAX}}(s_j) \quad (40)$$

$$OUT_{pi} = \Phi[\frac{1}{M}\sum_{j=1}^{M}\varepsilon_j] \quad (41)$$

$$\Phi(x) = \begin{cases} x & \text{if } x > \theta \\ 0 & \text{if } x \le \theta \end{cases} \quad (42)$$

where $M$ is the maximal number of nonzero Gaussian pairs in prototype $p_i$ and the input.

If $OUT_{p_i} > 0$

then prototype node $p_i$ is activated.

Step 2: If none of the prototype nodes in the $k^{th}$ category (i.e., the prototype nodes connected to the $k^{th}$ output node) is activated, then generate a new prototype node $p_i$ and connects it to the $k^{th}$ output node, and then set

$$a_i = [a_{i1}, a_{i2}, \ldots a_{i2M_{MAX}}] = x_0, b_{si} = b_{s0}, b_{ui} = b_{\mu 0}$$

, where $b_{s0}$ and $b_{\mu 0}$ are predefined values for all membership functions. An example is shown in Fig. 7.

Step 3: If any prototype node $p_i$ that does not connect to the $k^{th}$ output node is activated, then reduce $b_{si}$ and $b_{ui}$ as follows:

$$b_{si} = b_{si} - \Delta b_s, \text{ and } b_{ui} = b_{ui} - \Delta b_\mu \quad (43)$$

where $\Delta b_s$ and $\Delta b_\mu$ are predefined before training.

Therefore for each training cycle, when a prototype is activated incorrectly the widths of the membership functions will be reduced with a small, fixed value. Hence when repeatedly apply the same training input; smaller and smaller output from each membership function of the prototype will be obtained until the node is not activated. The neural spectrogram identifier retrieves as follows:

Step 1: For the input calculate the output ($OUT_{pi}$) of each prototype node ($p_i$).

Step 2: Find the prototype node with maximum output and assign its category to the input. Otherwise if none of the prototype node is activated, the identifier responses the input as undefined.

## 5. Experimental results

### 5.1 Acquiring the hyperspectral data

A color ink-jet printer (HP DeskJet 895Cxi) was used to print the sample image as shown in Fig. 8. Red, yellow and blue are the three basic colors. Mixing yellow and blue obtain light green and dark green. White regions on both sides help to locate the interested area. Black thin frame line of each color track shows the track boundaries. Fig. 9 shows the visual presentation of the spectra obtained by a hyperspectral imager for the blocks indicated in Fig. 8. The hyperspectral imager acquired the spectra of each pixel with 1024 different wavelengths, uniformly distributed from 473.0nm to 825.8nm [Wu 1999]. The spectral data taken from pixel number 40, 70, 125, 170 and 210, denoted as 'rygb1' to 'rygb5', respectively, were used to test the entire

procedure of identification.

Denote the reflectance spectrum of each pixel by

$$R(x_k), k = 1 \sim 1024 \quad (44)$$

where $x_k$ is the $k^{th}$ wavelength. Convert the reflectance spectrum into the apparent absorption spectrogram by

$$y_k = -\ln R(x_k), \quad k = 1 \sim 1024 \quad (45)$$

Remove the continuum by subtracting the minimum value of the apparent absorption spectrogram as

$$y'_k = y_k - \min_{k=1\sim1024} y_k \quad (46)$$

Applying the Gaussian Deconvolution to parameterize each spectrogram, the extracted Gaussian pairs of pixel number 40 (dark green), 70(blue), 125(light green), 170(yellow) and 210(red) are shown in Table 3. Table 4 shows the extracted Gaussian pairs of nearby pixels of rygb5_210. The results reveal that the numbers of the extracted Gaussian pairs are not the same. The Gaussian pairs ($1^{st}$, $2^{nd}$, $3^{rd}$ and last) with significant larger strengths appear in every pixel. This means significant absorption bands appear in all the sample pixels. Minor Gaussian pairs may be resulted by noise and provide no useful information for the identification. Therefore only the significant Gaussian pairs of each spectrogram as shown shadowed in Table 3 were selected for the identification.

5.2 Training the neural spectrogram identifier

For each category of the spectrograms, we select the nearby twelve pixels to form the sample data of this category. Two of the twelve sample pixels of each category were chosen as the training inputs. The rest ten samples of each category were mixed up and applied as unknown material to test the identification. Therefore there were fifty unknown samples for the trained identifier to discriminate. The initial values of the parameters used in the training procedure are listed in Table 5. After training, five prototype nodes were generated in the RCE neural network. Then applying the fifty unknown samples to the trained identifier, the results of identification are shown in Table 6. It works perfect except for Rygb3. The low accuracy in identifying Rygb3 (light green) is due to that four of its Gaussian pairs are close to that of Rygb1 (dark green), and the extra Gaussian pair of Rygb3 is not so significant for the discrimination. Conversely if the input is Rygb1, although it's four major Gaussian pairs are very close to that of Rygb3, respectively. But for being lack of a $5^{th}$ Gaussian pair, the outputs of the prototype nodes connected to Rygb3 may not be greater than the outputs of the prototype nodes connected to Rygb1. Thus the spectrograms of Rygb1 are seldom recognized as Rygb3.

5.3 Adding new categories to the trained neural spectrogram identifier

The hyperspectral data of five categories of objects were acquired by using the hyperspectral imager. The five categories include two kinds of rocks, a leaf, a T-shirt, and polystyrene. Fig. 11 shows the spectrograms of these five categories with the strength having been normalized to within the range 0~100. Thirteen samples were acquired from each category to give 5x13 samples in all. 5x3 samples of them were taken as training inputs, and the rest 5x10 samples were kept as unknown inputs to be identified.

These five new categories were applied to

train the spectrogram identifier that had been trained with Rygb1~5. After training with the 5x3 samples, the number of prototype nodes in the spectrogram identifier grew from five to fifteen. The rest 5x10 samples were then mixed up to test the identifier. The results of identification are shown in Table 7. The accurate result means the new training inputs do not corrupt those old information remembered in the identifier.

## 6. Conclusion

Hyperspectral imaging provides the potential to conduct remote identification or classification of surface compositions. Instead of working directly on the tremendous volumes of resultant data, the inputs of an identifier can be the quantitative information extracted from spectral parameters such as band wavelength positions, depths, widths, areas and absolute reflectance values. Assuming the absorption bands can be approximated by Gaussian distributions, this research developed the Gaussian Deconvolution to extract the band wavelength positions and strengths of the interested spectrograms as input features to the identifier. The proposed Gaussian Deconvolution algorithm is first using a filtered derivative analysis to locate approximately the wavelength positions of the absorption bands, then a constrained Gaussian function approximation is applied to estimate the strengths and widths of these bands, and finally using Gaussian function approximation refines all the Gaussian parameters. Examples have shown that this algorithm works successfully except the computation is heavy and time consuming. Identification is implemented by using a RCE neural network to learn to discriminate the Gaussian pairs extracted from the interested spectrograms and then retrieves the categories representing the spectrograms. The learning and retrieving procedures of the RCE neural identifier is efficient enough for real time applications. The simulation and experimental results show that the feature extraction by Gaussian Deconvolution and the identification based on the RCE neural network work effectively and accurately.


## Acknowledgment

The financial support for this research from the National Science Council of Taiwan, R. O. C. under NSC89-2212-E002-038 is gratefully acknowledged.


## References


[Ben-Dor 1994] E. Ben-Dor, and F. A. Kruse, 1994, Mineral mapping of Makhtesh Ramon Negev, Israel using GER 63 channel scanner data and three analyses methods. *Proceedings of the First Israel International Airborne Remote Sensing Conference and Exhibition*, Strasbourg, France, vol. III, pp 550-561.

[Blom 1980] R. G. Blom, M. J. Abrams, and H. J. Adams, 1980, Spectral reflectance and discrimination of plutonic rocks in the 0.45- to 2.45 μm region, *Journal of Geophysical Research*, vol. 85 pp. 2638-2648.

[Burden 1997] Richard L. Burden and J. Douglas Faires, 1997, *Numerical Analysis*, Brooks/Cole Publishing Company, p. 473-496, p. 599-604 and p. 607-610.

[Clark 1984] R. N. Clark and T. L. Roush 1984, Reflectance spectroscopy: quantitative analysis techniques for remote sensing applications, *Journal of Geophysical Research*,

vol. 89, no. B7, pp. 6329-6340.

[Cloutis 1996] E. A. Cloutis, 1996, Hyperspectral geological remote sensing: evaluation of analytical techniques, *International Journal of Remote Sensing*, vol. 17, no. 12, pp. 2215-2242.

[Crilly 1991] Paul Benjamin Crilly, 1991, A quantitative evaluation of various iterative deconvolution algorithms, *IEEE Trans. on Instrumentation and Measurement*, vol. 40, no. 3, pp. 558-562.

[Curtiss 1997] Curtiss O. Davisa and Kendall Carderb, 1997, Requirements driven design of an imaging spectrometer system for characterization of the coastal environment, *Proceedings of the SPIE* V. 3118, *San Diego, CA.*

[Elachi 1987] C. Elachi, 1987, *Introduction to the Physics and Techniques of Remote Sensing*, John Wiley & Sons, pp. 75-79.

[Huguenin 1986] R. L. Huguenin and J. L. Jones, 1986, Intelligent information extraction from reflectance spectroscopy: absorption band positions, *Journal of Geophysical Research*, vol. 91, no. B9, pp.9588-9598.

[Hunt 1978] G. R. Hunt and J. W. Salisbury, 1978, Assessment of Landsat filters for rock type discrimination, based on intrinsic information in laboratory spectra, Geophysics, vol. 43, pp. 738-747.

[Kawata 1984] S. Kawata and S. Minami, 1984, Adaptive Smoothing of Spectroscopic Data by A Linear Mean-Square Estimation, *Applied Spectroscopy*, vol. 38, no. 1, pp.49-58.

[King 1999] R. King and C. Ruffin, 1999, The analysis of Hyperspectral Data Using Savitzky-Golay Filtering-Theoretical Basis (Part 1) and Practical Issues (Part 2). *Geoscience and Remote Sensing Symposium*, vol. 1, pp.398-400, and vol. 2, pp. 756-758.

[Kruse 1989] F. A. Kruse, 1989, Spectral mapping with Landsat Thematic Mapper and imaging spectroscopy for precious metals exploration, *Proceedings of the 7th Thematic Conference on Remote Sensing for Exploration Geology*, Calgary, Alberta, pp. 17-78.

[Miekina 1997] Andrzej Miekina, Roman Z. Morawski and Andrzej Barwicz, 1997, The use of deconvolution and iterative optimization for spectrogram interpretation, *IEEE Trans. On Instrumentation and Measurement*, vol. 46, no. 4 pp. 1049-1053.

[Palmadesso 1995] P. Palmadesso, J. Antoniades, M. Baumback, J. Bowles and L. Rickard, 1995, Use of filter vectors and fast convex set methods in hyperspectral analysis, *Proceedings of the International Symposium on Spectral Sensing Research*, Melbourne, Australia.

[Reily 1982] D. L. Reily, L. N. Cooper, and C. Elbaum, 1982, A neural model for category learning, Biol. Cybern, vol. 45, pp. 35-41.

[Reily 1987] D. L. Reily, C. Elbaum and L. N. Cooper 1987, Learning system architectures composed of multiple learning modules, *Proceeding of IEEE Conference on Neural Networks*, vol. II, pp. 495.

[Roan 1993] J. A Richards 1993, *Remote Sensing Digital Image Analysis, 2nd ed.* Springer-Verlag, New York, pp. 181-226.

[Savitzky 1964] A. Savitzky and M. J. E. Golay, 1964, Smoothing and Differentiation of Data by Simplified Least Squares Procedures, *Analytical Chemistry*, vol. 36, no. 8, pp.1627-1639.

[Staenz 1996] K. Staenz, J. Schwarz and J. Cheriyan, 1996, Processing/analysis capabilities for data acquired with hyperspectral spaceborne sensors, *Acta Astronautica*, vol. 39, no. 9-12, pp. 923-931.

[Sunshine 1990] Sunshine, 1990, Deconvolution of mineral absorption bands: an improved approach, *Journal of Geophysical Res.* no. 95, pp. 6955-6966.

[Sunshine 1993] J.M. Sunshine and C.M. Pieters, 1993, Estimating modal abundance form the spectra of natural and laboratory pyroxene mixtures using the modified Gaussion model, *Journal of Geophysical Research*, vol.98, no.5, E5, pp.9075-9087.

[Tsai 1997] F. Tsai and W. Philpot, 1997, Derivative Analysis of Hyperspectral Data for Detecting Spectral Features, *Geoscience and Remote Sensing*, vol.3, pp.1243-1245.

[Wu 1999] Jackie Wu, 1999, Extraction of Absorption Features from Hyperspectral Data Using Gaussian Model with Parameter Initialization, thesis, Department of Electrical Engineering, National Taiwan University, Taiwan ROC.

Table 1 Parameters of Huguenin's test data and peak positions estimated by applying $f_1$ and $f_2$ filters with ($d$=10, $n$=20). Full-width-half-magnitude (FWHM) is denoted by $\omega_i = 2(2\ln 2)^{1/2}\sigma_i$.

| $g_i(x)$ | $\mu_i$ (nm) | $\omega_i$ (nm) | $s_i$ (%) | $\bar{\mu}_i$ (nm) | Err (%) |
|---|---|---|---|---|---|
| 1 | 9500 | 2355 | 30 | 9434 | 0.6990 |
| 2 | 11500 | 3040 | 42 | 11836 | 2.9212 |
| 3 | 14500 | 1990 | 30 | 14707 | 1.4278 |
| 4 | 16000 | 2150 | 34 | 15859 | 0.8789 |
| 5 | 18500 | 2033 | 60 | 18359 | 0.7601 |
| 6 | 20500 | 2150 | 80 | 20645 | 0.7050 |

Table 2 Results of the multi-step Gaussian functional approximation

| $w^{(0)}$ | | | $\bar{w}$ | | | $\hat{w}$ | | |
|---|---|---|---|---|---|---|---|---|
| $\bar{\mu}_i$ | $\omega_i^{(0)}$ | $s_i^{(0)}$ | $\bar{\mu}_i$ | $\bar{\omega}_i$ | $\bar{s}_i$ | $\hat{\mu}_i$ (Err %) | $\hat{\omega}_i$ (Err %) | $\hat{s}_i$ (Err %) |
| 9434 | 1.00 | 0.01 | 9434 | 715.00 | 57.81 | 9500 (0) | 2357.06 (0.09) | 30.00 (0) |
| 11836 | 1.00 | 0.01 | 11836 | 855.50 | 59.65 | 11500 (0) | 3042.66 (0.09) | 42.00 (0) |
| 14707 | 1.00 | 0.01 | 14707 | 822.60 | 57.91 | 14500 (0) | 1991.74 (0.09) | 30.00 (0) |
| 15859 | 1.00 | 0.01 | 15859 | 727.70 | 56.41 | 16000 (0) | 2151.88 (0.09) | 34.00 (0) |
| 18359 | 1.00 | 0.01 | 18359 | 1550.30 | 74.39 | 18500 (0) | 2034.78 (0.09) | 60.00 (0) |
| 20645 | 1.00 | 0.01 | 20645 | 1786.50 | 79.24 | 20500 (0) | 2151.88 (0.09) | 80.00 (0) |

| Derivative analysis | Steepest descent rule | The Hessian's iterative rule |
|---|---|---|
| Using $f_1$, $f_2$ with | 52 iterations | 7 iterations |
| $(d,n)=(10,20)$ | $\delta^{(j)}=0.05$ =constant | |
| $\phi(w^{(0)})=1.7719\times10^{6}$ | $\phi(w^{(52)})=3.4480\times10^{5}$ | $\phi(w^{(59)})=7.8887\times10^{-7}$ |

Table 3 The extracted Gaussian pairs of pixels 40, 70, 125, 170 and 210.

| Gaussian | 1st | 2nd | 3rd | 4th | 5th | 6th | 7th | |
|---|---|---|---|---|---|---|---|---|
| Rygb1 | | | | | 871.14 | | | Peak |
| | | | | | 8.74 | | | Strength |
| Rygb2 | | 508.18 | | | | 2156.8 | | Peak |
| | | 0.31 | | | | 18.43 | | Strength |
| Rygb3 | | | | | | | | Peak |
| | | | | | | | | Strength |
| Rygb4 | | | | 790.52 | 426.44 | 885.73 | | Peak |
| | | | | 4.20 | 0.26 | 0.56 | | Strength |
| Rygb5 | | | | 689.49 | 733.30 | | | Peak |
| | | | | 1.20 | 2.83 | | | Strength |

Table 4 The extracted Gaussain pairs of the red color bar (rygb5)

| Gaussain | 1st | 2nd | 3rd | 4th | 5th | 6th | 7th | |
|---|---|---|---|---|---|---|---|---|
| Red_210 | | | | 689.49 | 733.30 | | | Peak |
| | | | | 1.20 | 2.83 | | | Strength |
| Red_211 | | | | 792.98 | 840.50 | 727.47 | | Peak |
| | | | | 2.71 | 1.73 | 0.66 | | Strength |
| Red_212 | | | | 890.20 | 751.37 | | | Peak |
| | | | | 3.48 | 0.17 | | | Strength |
| Red_213 | | | | 725.38 | 826.93 | 709.54 | | Peak |
| | | | | 0.81 | 1.46 | 0.25 | | Strength |
| Red_214 | | | | 797.30 | 592.39 | | | Peak |
| | | | | 3.48 | 0.09 | | | Strength |
| Red_215 | | | | 989.62 | 810.23 | 733.82 | | Peak |
| | | | | 0.93 | 0.24 | 1.56 | | Strength |

Table 5 Initial values of the training parameters

| $\theta$ | $b_{s0}$ | $b_{\mu 0}$ | $\Delta b_s$ | $\Delta b_\mu$ | $w_s$ | $w_\mu$ |
|---|---|---|---|---|---|---|
| 0.6 | 8 | 40 | 1 | 5 | 0.35 | 0.65 |

Table 6 Identification result of the Rygb color bar

| Category | Rygb1 | Rygb2 | Rygb3 | Rygb4 | Rygb5 | Total |
|---|---|---|---|---|---|---|
| Samples | 10 | 10 | 10 | 10 | 10 | 50 |
| Accuracy | 100% | 100% | 80% | 100% | 100% | 96% |

Table 7 Identification results of the five new categories

| Category | Rock 1 | Rock 2 | Leaf | Polystyrene | T-shirt |
|---|---|---|---|---|---|
| Accuracy | 80% | 90% | 100% | 90% | 100% |

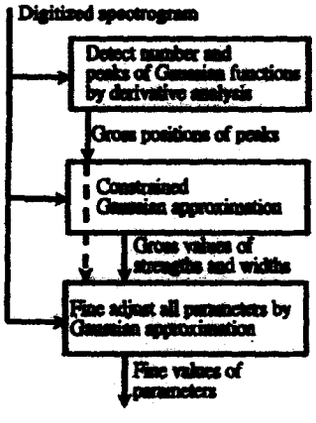


Fig.1 A multi-step Gaussian approximation.

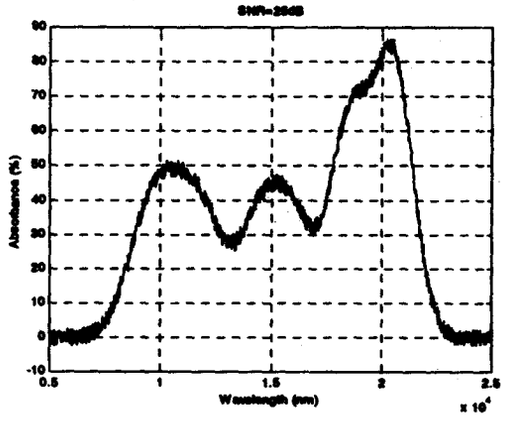

Fig. 2 Huguenin's test data corrupted with white noise, SNR=25dB.

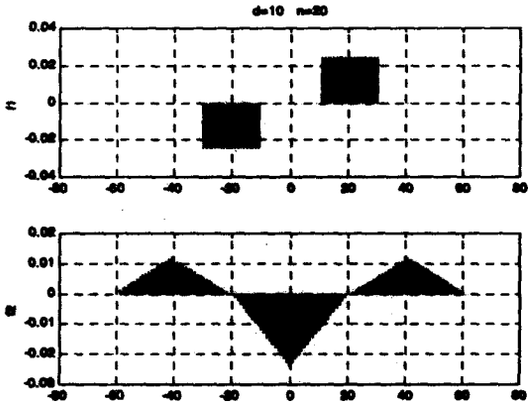

Fig. 3 Impulse responses of $f_1$ and $f_2$, ($d$, $n$)= (10, 20).

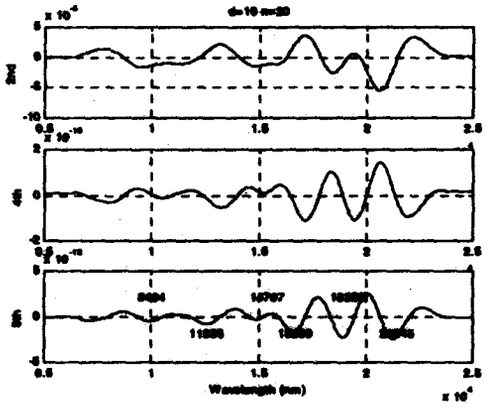

Fig. 4 The three derivatives and peak positions obtained by applying the $f_1$ and $f_2$ filters ($d$=10, $n$=20).

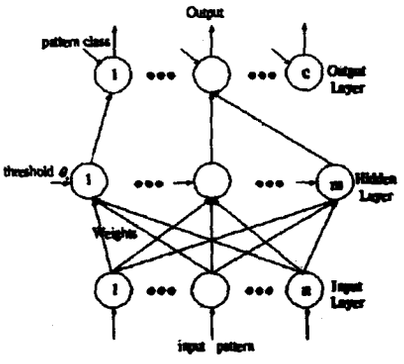


Fig. 5 The architecture of a basic RCE neural network.

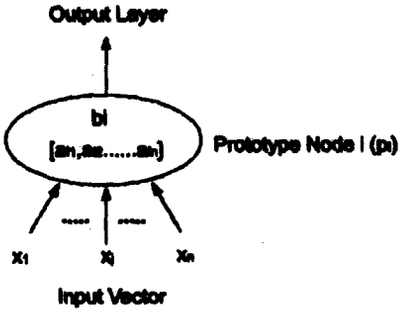


Fig. 6 A prototype neuron in the RCE network.

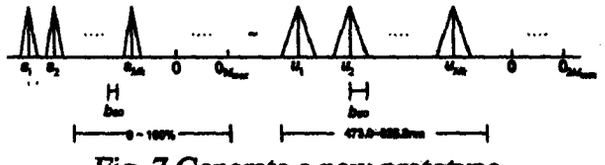

Fig. 7 Generate a new prototype.

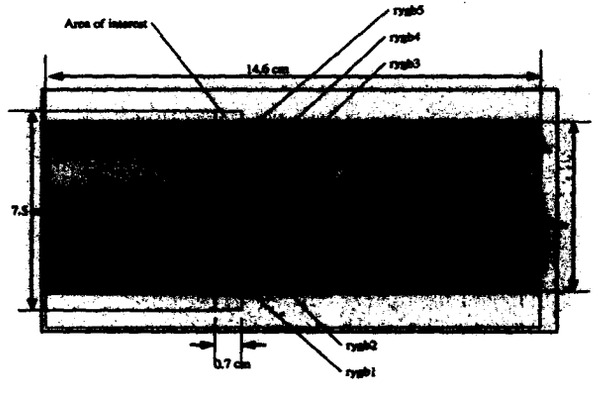


Fig. 8 A printed sample image.


Fig. 9 The hyperspectral image obtained from the sampled blocks indicated in Fig. 8.

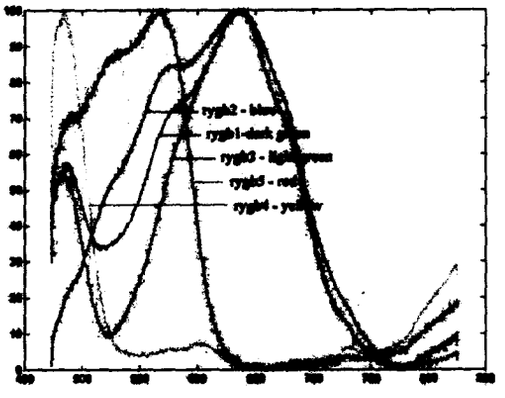

Fig. 10 The five spectrograms shown in apparent absorbance.

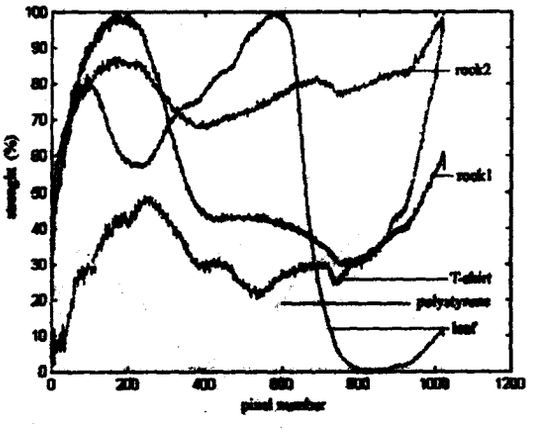


Fig. 11 Spectra of the two rocks, leaf, polystyrene and T-shirt.