The 6th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications
15-17 September 2011, Prague, Czech Republic

Time in Hyperspectral Processing: a Temporal based Classification Approach

Blanca Priego, Daniel Souto, Francisco Bellas, Richard J. Duro and Fernando López-Peña
Grupo Integrado de Ingeniería, Universidade da Coruña
Mendizábal s.n., 15403 Ferrol, Spain
Email: {blanca.priego, dsouto, fran, richard, flop}@udc.es

Abstract—**This paper deals with the problem of classifying processes using the temporal information in the sequence of hyperspectral images that are obtained as they take place. That is, taking into account the temporal evolution of the process in the discrimination that must be made. To this end we have considered a particular type of artificial neural networks with trainable delays in their synapses. The classification scheme is studied and applied to the case of resin curing processes. Several test cases involving different proportions of resin components as well as certain environmental conditions such as humidity were created and the system was tested over them producing very promising results.**

Keywords—Temporal Delay Based Artificial Neural Netorks; Hyperspectral imaging;Real Time processing.

I. INTRODUCTION

Hyperspectral imaging has traditionally been used in remote sensing instruments. The first hyperspectrometers were developed for imaging from high flying airborne platforms or satellites. They were usually large instruments with complex deployment and handling characteristics typically run by space agencies or other large providers of imaging resources. However, due to the popularization of imaging sensors and the advances in digital photography and video capture technology, in the last ten years we have seen the implementation of many smaller designs and platforms that have opened up new application domains, especially in close up inspection tasks such as medical imaging or quality control in processing plants, leading to a flurry of activity in research into new algorithms and strategies that would adapt to these new application areas.

In hyperspectral imaging systems, for each spatial resolution element (pixel), data is collected with high spectral resolution over the electromagnetic spectrum in the 400-2500 nm band (visible to near infrared). It is commonplace to use 50 to 250 spectral bands with bandwidths in the 5 to 20 nm range. The large amount of information that any hyperspectral image provides permits a detailed description of the spectral signature for each pixel in the image, thus greatly improving the ability to detect and identify individual materials or classes with respect to other remote sensing techniques. Unfortunately, on the other hand, these large amounts of data are the source of some of the main challenges currently associated with hyperspectral imaging: efficient handling of high data rates and accurate and fast segmentation of images. Hyperspectrometric systems represent today a mature technology, and have been applied to many different instances. Typical applications of remote hyperspectral sensing are related to vegetation monitoring [1]-[3], target detection [4], [5], and many others. In addition to remote sensing applications, hyperspectral imaging is also being used in close up inspection tasks such as medical imaging or processing plant visual inspection [6], [7].

There is presently a wide open field for new applications using hyperspectral imaging at close and mid-range that would make these instruments much more accessible and ubiquitous. However, it does require an impulse in two main directions for the technology to become commonplace and popular. On one hand, hyperspectrometers must be made more affordable, small, light and rugged and, on the other, they must be made as autonomous as possible and very easy to use by non-experts. This obviously requires developments in the hardware and control structure and motivates the research into efficient and accurate methods to process this data. In particular, as autonomy and data processing resilience would be a very important aspect of this new generation of sensors, the introduction of more intensive computational intelligence techniques that allow the systems to be used in less specialized applications than those currently contemplated becomes a necessity.

According to Manolakis et al [8], most algorithms used in hyperspectral applications can be grouped into four categories: change detection, target/anomaly detection, classification, and spectral unmixing. Up to now, almost all research efforts have focused on the analysis of static hyperspectral images in terms of the last three categories. However, in the last few years, as a consequence of improvements in hyperspectrometers, it is becoming feasible to capture a continuous flux of images allowing one to look at hyperspectral information dynamically. Some algorithms have been designed using temporal information [9] [10], mostly for the detection of changes, but still without really using the evolution seen in the frame sequence as a classification strategy. Here, our objective is to formulate techniques that permit addressing the problem of using temporal sequences in the classification process and not only to detect changes. In fact, there are many problems where information from a single image provides ambiguous classification results and it is the integration of the evolution of the subject in time that really provides an unambiguous classification.

This problem has been tackled in the image processing field dealing with one or three dimensional (RGB) images by using video sequences to obtain better classifications, to eliminate noise or to improve image resolution. Here the objective is to extend this work to the realm of high dimensional images, such as those obtained from hyperspectral sensors, through the use of new techniques and algorithms that are adapted to the high dimensionality involved. In particular, in this paper we make use of including specific types of neural network architectures and training procedures for the temporal processing of this type of images.

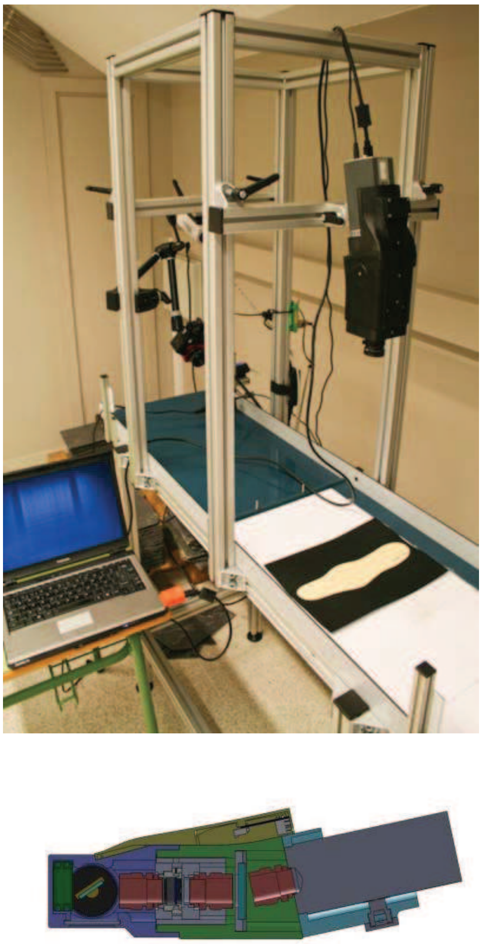

Figure 1. Experimental setup (top) and schematic view of the hyperspectrometer developed

II. SENSING SYSTEM

The sensor used in this work has been developed by our group as a small, light and easy to deploy hyperspectral sensor that was designed to explore the 400 to 1000 nm band with a spectral resolution of up to 1040 bands. Figure 1 displays the experimental setup that was used and a schematic view of the hyperspectrometer. This sensor is capable of capturing up to 47.2 hyperspectral 1040 pixel lines per second with the spectral depth indicated before.

III. TEMPORAL BASED PIXEL CLASSIFICATION

To address the problem of performing classifications taking into account the temporal evolution of the spectra of the pixels, we have chosen to use a neural network architecture developed in our group that is called Temporal Delay Based ANN as well as its training algorithm (TDBP). The architecture and training algorithm of the artificial neural network we consider were introduced in [11], [12]. The network consists of several layers of neurons connected like a Multiple Layer Perceptron (MLP), that is, every neuron of one layer is connected through a synapsis to every neuron of the next layer. Each neuron performs a sum of its inputs and passes these values through some non-linear function (in this case a sigmoid). It is obviously a feed-forward network. The only difference with respect to a traditional MLP is that the synapses are represented by two trainable parameters: the classical weight term and a delay term. Consequently, now the synaptic connections between neurons are characterized by a pair of values, (w_{ij}, τ_{ij}), where w_{ij} is the weight describing the ability of the synapsis to transmit information from neuron *i* to neuron *j* and τ_{ij} is a delay, which can be taken as an indication of the length of the synapsis between neurons *i* and *j*, the longer it is, it will take more time for information to traverse it and reach the target neuron.

The training algorithm is described in detail in [11] where it was used on one-dimensional signals. Here we will just provide a summary of its main points and how it is used on multidimensional signals.

The main assumption during training in this algorithm is that each neuron in a given layer can choose the delay it wishes to impose on its inputs. Time is discretized into instants, each one of which corresponds to the period of time between an input to the network and the next input. Every neuron of the network computes an output each instant of time.

In order to choose from the possible inputs to a neuron the ones we are actually going to take as input it in a given instant of time, we add a selection function to the processing of the neuron. This selection function can be something as simple as:

$$\delta_{ij} = \begin{cases} 1 \rightarrow i = j \\ 0 \rightarrow i \neq j \end{cases} \qquad (1)$$

Making output of neuron k in instant t:

$$O_{kt} = F\left(\sum_{i=0}^{N} \sum_{j=0}^{t} \delta_{j(t-\tau_{ik})} w_{ik} h_{ij} \right) \quad (2)$$

where F is the activation function of the neuron, h_{ij} is the output of neuron i of the previous layer in instant j and w_{ik} is the weight of the synapsis between neuron i and neuron k. The first sum is over all the neurons that reach neuron k (those of the previous layer) and the second one is over all the instants of time considered.

The result of this function is the sum of the outputs of the hidden neurons in times t-τ_{jk} (where τ_{jk} is the delay in the synapsis) weighed by the corresponding weight values.

This determines the output of every neuron as a function of the outputs of the neurons in the previous layer and the weights and delays in the synapses. To train these weights and delays we have resorted to a modification of the basic gradient descent algorithm employed in traditional backpropagation, taking into account the delay terms when computing the gradients of the error with respect to weights and delays. Thus, the gradient terms can be written as:

$$\frac{\partial E_{total}}{\partial w_{jk}} = \Delta_k h_{j(t-\tau_{jk})} \quad (3)$$

$$\frac{\partial E_{total}}{\partial \tau_{jk}} = \Delta_k w_{jk} \left(h_{j(t-\tau_{jk})} - h_{j(t-\tau_{jk}-1)} \right) \quad (4)$$

in which E_{total} is the total squared error for all the training vectors and

$$\Delta_k = \frac{\partial E_{total}}{\partial O_k} \frac{\partial O_k}{\partial ONet_k} = 2(O_k - T_k) F'(ONet_k) \quad (5)$$

where T_k is the target output, O_k the one really obtained and $ONet_k$ is the combination of inputs to neuron k, when we consider output neurons and:

$$\frac{\partial E_{total}}{\partial w_{jk}} = \frac{\partial E_{total}}{\partial hNet_k} \frac{\partial hNet_k}{\partial w_{jk}} = I_{j(t-\tau_{jk})} F'(hNet_k) \sum_r \Delta_r w_{kr} \quad (6)$$

$$\frac{\partial E_{total}}{\partial \tau_{jk}} = \frac{\partial E_{total}}{\partial hNet_k} \frac{\partial hNet_k}{\partial \tau_{jk}} = w_{jk} \left(I_{j(t-\tau_{jk})} - I_{j(t-\tau_{jk}-1)} \right) F'(hNet_k) \sum_r \Delta_r w_{kr} \quad (7)$$

where index r represents the neuron of the next layer, whether output or hidden. We assume:

$$\Delta_k = \frac{\partial E_{total}}{\partial hNet_k} = F'(hNet_k) \sum_r \Delta_r w_{kr} \quad (8)$$

and

$$\frac{\partial hNet_k}{\partial \tau_{jk}} = \frac{\partial \left[\sum_{i=0}^{N} \sum_{n=0}^{t} \delta_{n(t-\tau_{i,k})} w_{ik} I_{in} \right]}{\partial \tau_{jk}} \quad (9)$$

for the case of neurons in a hidden layer.

Assuming there is certain continuity in the temporal variation of the outputs of the neurons, the derivative in *hNet* of equation (9) has been discretized in order to obtain (7).

Summarizing, by discretizing the time derivative simple expressions may be obtained for the modification of the weights and delays of the synapses in an algorithm that is basically a backpropagation algorithm where the activation function of the neuron has been modified to permit selecting delays or, in order words, choosing from the list of previous outputs of the neuron en the previous layer. By adding input neurons to the network, any dimensionality of the signals can be chosen.

IV. Experimental Results

The objective of this paper is to demonstrate how by considering the temporal sequence of hyperspectral images taken during the evolution or a process and appropriately processing them we can glean information that is hard or impossible to obtain using static images. To test this hypothesis we have performed a series of experiments related to quality control of the drying or curing processes after applying different surface coatings. The basic idea is to determine if the resulting surface coating meets the quality levels as a function of the temporal drying sequence. To this end, we make use of the sequence of hyperspectral images of the products as time progresses. To obtain adequately cured resins it is necessary to mix certain components (in this case two) in the appropriate proportions and let the resin cure for a period of time under adequate environmental conditions. In fact, this is really a spatial-temporal problem as the classification may be different for different areas of the resin surface as shown in figure 2. In the end, it is the determination of whether the resin has cured appropriately in every point that is important. This is where the use of hyperspectral imaging makes sense.

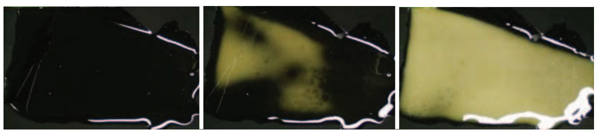

Figure 2. Three instants in the evolution of the drying process as obtained by composing bands from the visible spectra.

Several different mixtures were made under different humidity conditions and with different proportions of components. One of them, which will be called "correct curing process" in the graphics and comments that will follow, has been taken as the reference mixture and conditions for which the curing process is optimal. The remaining cases were considered suboptimal. Table 1 provides a brief description of the different experiments.

Using the hyperspectrometric setup described before, the mixtures were made and left to cure for 500 seconds. Images were taken by the hyperspectrometer of a line of 1000 points traversing the mixtures at a rate of close to one image per second (485 images in 500 seconds). The spectra corresponding to the points obtained were normalized and corrected using as a reference the background area where there was no mixture. The objective was for the processing system to discriminate the optimal curing processes from suboptimal ones as soon as possible. Figure 2 presents the visual appearance of the curing process as it progresses.

Table I. CURING PROCESSES

EXPERIMENT 1 Different component percentage. Mixing perfomed by person X	
1.1	50% A, 50% B
1.2	20% A, 80% B
1.3	30% A, 70% B
1.4	40% A, 60% B
1.5	45% A, 55% B
1.6	55% A, 45% B
1.7	60% A, 30% B
EXPERIMENT 2 Humid Conditions	
2.1	Very wet curing container
2.2	Slightly wet curing container
2.3	5%Water, 47.5% A, 47.5% B
2.4	1%Water, 49.5% A, 49.5% B
EXPERIMENT 3	
	Non-consistent, irregular stirring motion
EXPERIMENT 4 Different component percentage. Mixing performed by person Y	
4.1	40%A, 60%B
4.2	45%A, 55%B
4.3	55%A, 45%B
4.4	60%A, 40%B
EXPERIMENT 5	
	Two different mixtures mixed with slight stirring Mixing 1: 50%A, 50%B Mixing 2: 40%A, 60%B

As a first test in order to validate the need to perform a time based processing, we evaluated the possibility of discriminating the correctly cured resin from the rest once the whole curing process was finished. Figure 3 displays the spectra corresponding to points on three different experiments once all of them had cured. As shown, the spectra are practically identical and it was impossible to obtain any kind of correct classification using only this information.

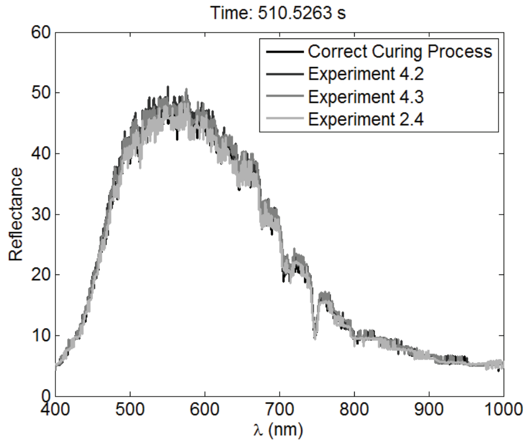

Figure 3. Spectra of the final resins after curing has ended for different cases (a correct and three incorrect curing processes).

As a consequence, we considered the temporal evolution and introduced the delay based neural networks as classifiers. Figure 4 displays a sequence of spectra for two cases, one the correct curing process and another one that is incorrect.

To train these networks we used 80% of the samples obtained as described above. The remaining 20% were used as test samples. For each experiment, the temporal sequence was discretized into ten time intervals by averaging the spectra within each interval and the spectra themselves were binned into 16 bands. A 16 input TDANN was used and different proportions of the temporal sequence were used for training and testing in order to determine how much temporal information from the curing process was necessary in order to achieve 100% discrimination (in this case positive classification would return a value of 0.5 and a negative of -0.5). This process was started using the first 36 seconds of curing and, as shown in figure 5, even though with this information a certain order in terms of quality seems to be present, the discrimination is still not good enough and there is a certain level of confusion. This level decreases if more time is considered as shown in the graphs of figure 6, corresponding to 42, 47, 53 and 480 seconds of sampling. In fact, in cases with more than 47 seconds of sampling, all the correctly cured samples (blue dots) are appropriately classified with a value above zero and all the rest of incorrect curing processes are assigned values below zero by the network. Consequently, by monitoring the processes for fifty seconds we are able to discriminate perfectly if it is going to cure correctly or not. Obviously, if more time is considered, the results become even better as shown in the bottom graph corresponding to the results produced by the network considering 48 seconds, that is, almost the whole 500 second time interval.

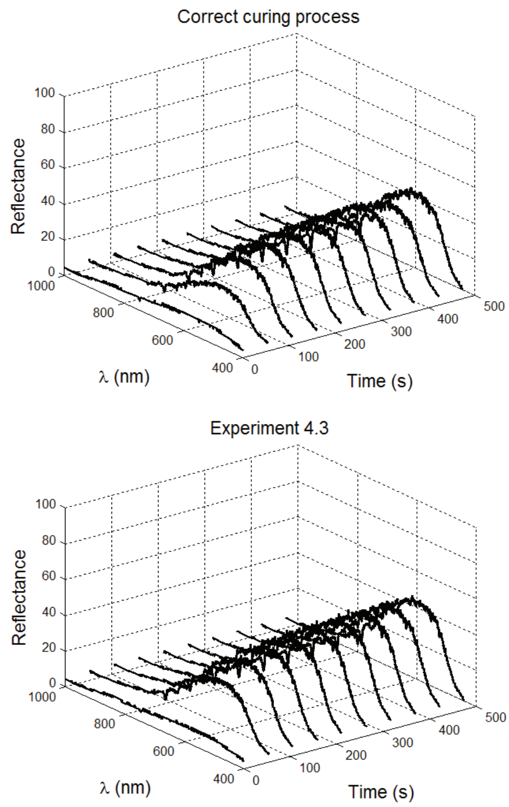

Figure 4. Evolution of the spectra during the curing process for the correct one (top) and that of experiment 4.3 (bottom).

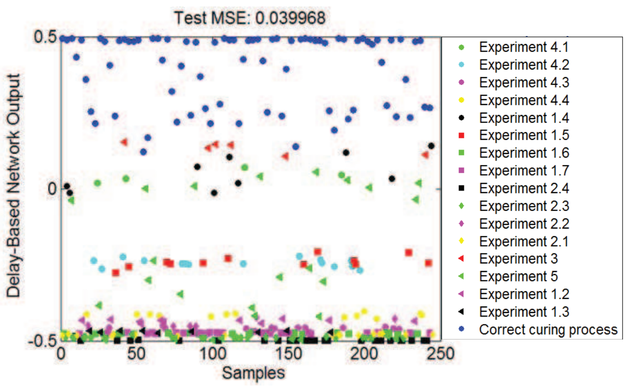

Figure 5. Results produced by the TDANN when using a sequence of 10 averaged spectra from the first 36 seconds of the curing process.

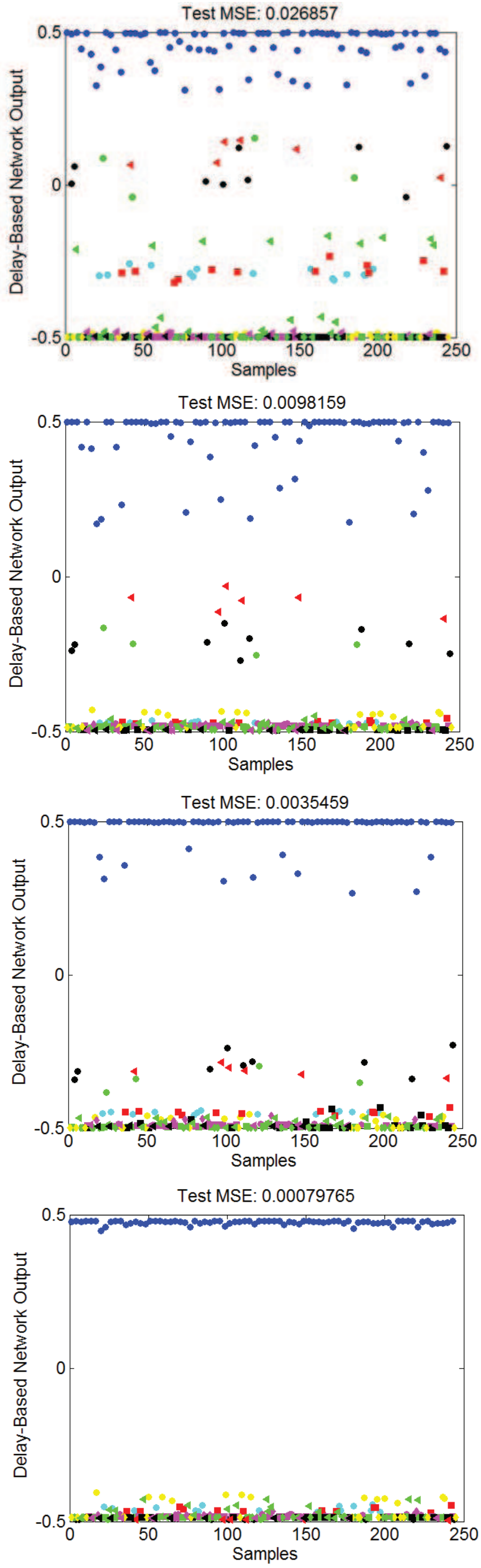

Figure 6. Same as figure 5 (including process labels) but taking into account, from top to bottom, 42, 47, 53 and 480 seconds of curing process.

Finally, and to demonstrate how fast this algorithm can learn, in Figure 7 we display the evolution of the Mean Squared Error as training takes place for some of the previous cases. The algorithm only requires around 150 epochs of training to achieve very low errors in those cases where the temporal information is enough to establish a reliable discrimination. In fact, this number decreases to around 70 epochs in the case of the 480 second sampling.

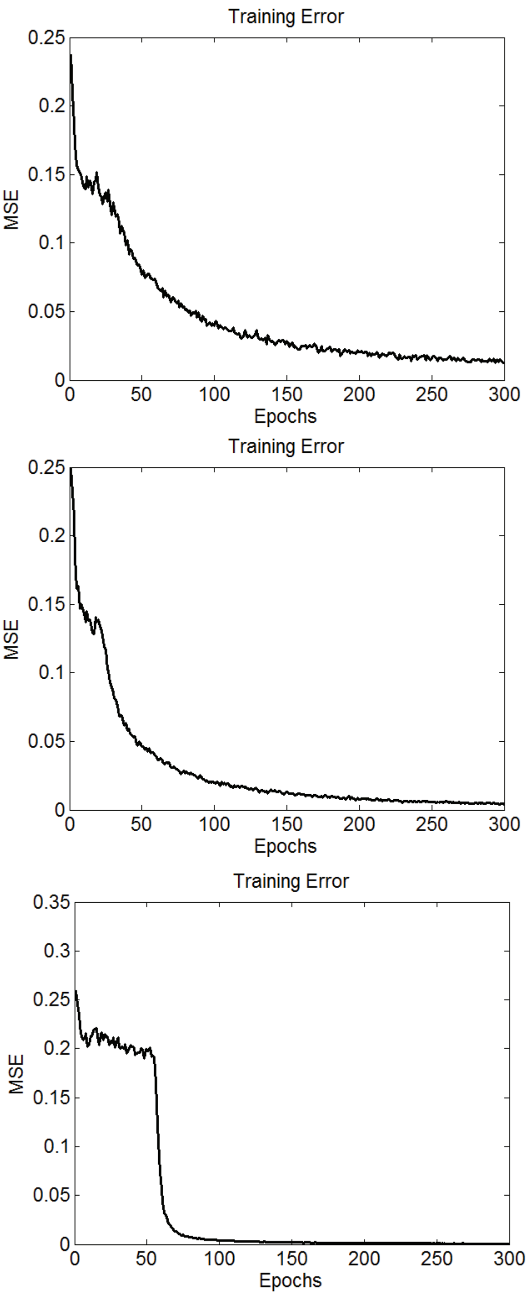

Figure 7. Training Error for the training processes using 47, 53 and 480 seconds of sampling

V. CONCLUSIONS

In this paper we have shown how the temporal evolution of the spectra within hyperspectral images can be used in order to classify the quality of processes even when the final spectra are almost the same for correctly performed processes or processes with deficiencies. To this end we have considered the application of synaptic delay based artificial neural networks to the task.

It has been shown that this type of approach can discriminate very clearly between correctly performed curing processes and processes that present problems due to wrong component mixtures or humidity. In fact, this discrimination can be performed using just the first few seconds of the curing process.

We are now in the process of extending these results to other processes in what we think is a very promising approach to expand the use of hyperspectral imaging.

ACKNOWLEDGMENT

This work was partially funded by the Xunta de Galicia and European Regional Development Funds through projects 09DPI012166PR and 10DPI005CT.

REFERENCES

[1] Jens Oldeland, et al. “Combining vegetation indices, constrained ordination and fuzzy classification for mapping semi-natural vegetation units from hyperspectral imagery,” *Remote Sensing of Environment*, 114(6):1155–1166, June 2010.

[2] V. Heikkinen, T. Tokola, J. Parkkinen, I. Korpela, and T. Jaaskelainen. “Simulated multispectral imagery for tree species classification using support vector machines,” *Geosci. Rem. Sens., IEEE Trans.*, 48(3):1355–1364, 2010.

[3] G.H. Mitri and I.Z. Gitas. “Mapping postfire vegetation recovery using EO- 1 hyperion imagery,” *Geosci. Rem. Sens., IEEE Trans.*, 48(3):1613–1618, 2010.

[4] D.E. Bar, K. Wolowelsky, Y. Swirski, Z. Figov, A. Michaeli, Y. Vaynzof, Y. Abramovitz, A. Ben-Dov, O. Yaron, L. Weizman, and R. Adar. “Target detection and verification via airborne hyperspectral and High-Resolution imagery processing and fusion,” *Sensors Journal, IEEE,* 10(3):707–711, 2010.

[5] J. Broadwater and R. Chellappa. “Hybrid detectors for subpixel targets,” *Pattern Analysis and Machine Intelligence, IEEE Trans.*, 29(11):1891–1903, 2007.

[6] Hiroshi Okamoto and Won Suk Lee. “Green citrus detection using hyperspectral imaging,” *Computers and Electronics in Agriculture*, 66(2):201–208, May 2009.

[7] L. Liu, M.O. Ngadi, et al. “Categorization of pork quality using gabor filter-based hyperspectral imaging technology,” *Journal of Food Engineering*, 99(3):284–293, August 2010.

[8] D. Manolakis, C. Siracusa and G. Shaw, “Hyperspectral subpixel target detection using the linear mixing model,” *IEEE Trans. Geoscience and Remote Sensing*, vol. 39, pp. 1392-1409, 2001.

[9] Murzina, Marina V.A. and Farrell, J. Paul. “Dynamic Hyperspectral Imaging.” *Nondestructive Detection*, vol. 5679, pp. 135-144. 2005.

[10] Blackburn, Joshua, “Feature Aided Tracking with Hyperspectral Imagery,” *Signal and Data Processing of Small Targets*, vol. 6699S, 2007.

[11] Duro, R.J., Santos J. “Discrete Time Backpropagation for training Synaptic Delay Based Artificial Neural Networks,” *IEEE Transactions on Neural Networks*. vol.10, pp. 779-789. 1999

[12] Santos, J. and Duro, R.J. “Influence of Noise on Discrete Time Backpropagation Trained Networks,” *Neurocomputing*. vol. 41, pp 67-89. (2001).