Optical Flame Detection Using Large-Scale Artificial Neural Networks

Javid Huseynov[†*], Zvi Boger[‡], Gary Shubinsky[*], Shankar Baliga[*]
javid@ics.uci.edu, zvi@peeron.com, garys@generalmonitors.com, shankarb@generalmonitors.com

[†] School of Information and Computer Science, University of California Irvine, Irvine, CA 92697
[‡] OPTIMAL – Industrial Neural Systems, Ltd., Be'er Sheva 84243, Israel and
OPTIMAL Neural Informatics, LLC, Rockville, MD 20852
[*] General Monitors, Inc., 26776 Simpatica Circle, Lake Forest, CA 92630

April 4, 2005

***Abstract* – A model for intelligent hydrocarbon flame detection using artificial neural networks (ANN) with a large number of inputs is presented. Joint time-frequency analysis in the form of Short-Time Fourier Transform was used for extracting the relevant features from infrared sensor signals. After appropriate scaling, this information was provided as an input for the ANN training algorithm based on conjugate-gradient (CG) descent method. A classification scheme with trained ANN connection weights was implemented on a digital signal processor for an industrial hydrocarbon flame detector.**

I. INTRODUCTION

Infrared (IR) optical sensors are broadly used in industrial hydrocarbon flame detection. Their popularity is dictated by the fixed emission wavelengths of hydrocarbon flames in the IR spectrum, which can be separated from most non-flame sources and analyzed in various domains. Classical optical hydrocarbon flame detectors are based on an expert system, where analog signals are collected from the optical sensors, converted into digital format, processed, and an output decision is reported on the presence of flame or lack thereof.

Although simple in appearance, the described model of flame detection becomes more complex when dealing with IR data from real industrial environments. IR signals at flame wavelengths can be easily generated by a random motion, modulation of heated surfaces, hot air flow, arc welding, reflection off water surface, and other non-flame related environmental nuisance.

Optical flame detection manufacturers have attempted to resolve this problem by using multiple sensors, each at a different wavelength [1-3]. In addition to wavelength discrimination via use of multiple sensors, most optical detectors measure the temporal characteristics of the signal, thereby analyzing the flame flickering properties [1]. Various signal-processing techniques such as correlation, taking ratios, frequency analysis, periodicity check, and threshold crossing are used in industrial flame detection to discriminate flames from non-flames.

The apparent difficulty of *linear* separation of flames from non-flame sources drives the usage of more sensors at a variety of wavelengths. In practice, this solution is very laborious and difficult to implement as an expert system. So, there arises an interest in *non-linear* pattern recognition methods, in particular, artificial neural networks.

Artificial neural network (ANN) is used for analyzing data when mathematical relationships between the inputs and the outputs of a system are not easily derivable. The application of ANN in a safety-driven environment, such as a gas plant or an oil refinery, requires a thorough understanding of the environmental characteristics that should be learned and classified. This understanding is pivotal in the design of a feature extraction scheme for ANN.

Previously, researchers have attempted to use multiple sensors along with advanced artificial intelligence methods to build a new generation of fire detection systems [4-6]. To our knowledge, our design is the first in applying ANN with over 1000 input features along with advanced signal pre-processing to design a detection algorithm for an industrial IR flame detector [7].

II. FEATURE EXTRACTION

In any IR flame detector, proper differentiation of flame sources from non-flame sources requires an elaborate signal-processing scheme for extracting relevant patterns from the input signal. Typically, signals are analyzed in time and/or frequency domain. But there are two important questions to be answered before designing a signal-processing algorithm for an embedded system:

1) Is the input signal analyzed in time domain or frequency domain, or both?

Analyzing sensor response in time domain only is complex due to variance of timed signal patterns relevant to the same environmental phenomenon. For instance, signals produced by the same flame source can vary in amplitude and function depending on distance, angle, presence of obstacles and other non-flame conditions (sunlight, wind, random modulation, bright lights, rain, fog, dust), which may look like flame to an IR sensor.

As opposed to time domain, the frequency domain of flame flickering remains relatively independent of the environmental conditions, so frequency becomes an important parameter for analyzing signals. But due to the low-frequency range of flame flickering, the frequency-only analysis is prone to low-frequency noise from non-flame sources, and time domain information is needed.

To avoid the drawbacks of time-only and frequency-only signal processing methods, joint time-frequency analysis (JTFA) is used for precise tracking of the frequencies of non-stationary time-varying signals [8].

2) What is the scale and multiplicity of input patterns that can be recognized within the limitations of an embedded processor?

In an expert system, the amount of information that can be extracted from an input signal is limited by the design of a static pattern-matching algorithm. Obviously, with the processing and memory limitations of an embedded stand-alone system, the number of expert-encoded patterns to compare to is also limited.

On the other hand, an ANN-based intelligent system learns from unlimited number of patterns outside the embedded system, on a separate workstation [7]. A fixed-size set of trained connection weights is loaded into the embedded system for use in classification. This way, the quality of classification is not limited by the complexity of the algorithm or the embedded system resources but depends on the quality of input data and the ANN training.

A. Short-Time Fourier Transform (STFT)

Fast Fourier Transform (FFT) is a classical method for extracting frequency information from the time-invariant input signals. However, for quasi-stationary signals, such as those of speech, music, video, IR source, computing the spectrum of the complete signal, from $-\infty$ to $+\infty$, will make it difficult to extract distinctive frequency information that changes over time [9]. In an effort to combine the Fourier analysis with time domain information, Dennis Gabor (1946) adapted the Fourier Transform to analyze only a small section of the signal at a time [10]. This adaptation is known as Short-Time Fourier Transform (STFT), or simply as Gabor Transform.

In STFT application, the input signal is cut into slices, followed by application of FFT to individual slices. The functions obtained by such segmentation are not periodic, which results in high Fourier coefficients at high frequencies, since FFT interprets jumps between slices as abrupt changes in signal. Such *spectral leakage* [9] is resolved by application of *data windowing*, when an input signal buffer is multiplied by a raised cosine wave.

The mathematical formulation of STFT is given below:

$$X_l(k) = \sum_{n=0}^{N-1} w(n)x(n+lH)e^{-j\omega_k n},$$

where n is the number of time samples, $w(n)$ is a data window, H is the window shift size, $x(n)$ is the input signal, $\omega_k = 2\pi f_k / f_s$ is the frequency f_k of the k^{th} Fourier transform bin, normalized by sampling frequency f_s, and $l = 0,1,2,\dots$ is a discrete frame.

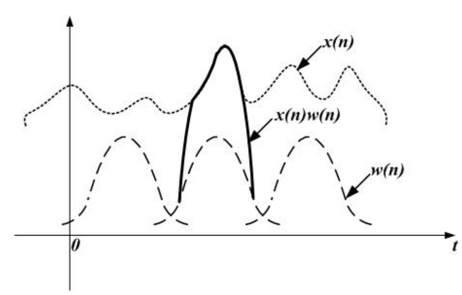

Figure 1. Application of STFT with data windowing.

B. Data Windowing.

Data windowing gradually attenuates the amplitude of signal at either end of the input buffer, hence reduces the spectral leakage into adjacent slices and forces the input wave to be more periodic. There are several known functions for data windowing such as Hamming, Hanning, Parzen, Parzen, Gaussian, etc. In our application, we have experimentally identified the application of the Hamming window to result in the best final ANN classification. The functional representation of the Hamming window is as follows:

$$W^{Hm}(n) = \frac{1}{2}\left\{1.08 - 0.92\cos\left(\frac{2\pi n}{N-1}\right)\right\},$$

where N is the size of the window, and n is the variable index.

Besides STFT, the Discrete Wavelet Transform (DWT) [11] can be used for signal pre-processing with equal or better ANN classification results.

III. ANN Model

The objective of ANN-based classification is to establish numerical representations of input-output relationships without a priori knowledge of system structure. Our ANN algorithm uses conjugate-gradient (CG) descent method for feed-forward networks [12]. The CG method is ideal for training multilayer neural networks with a large (over 100) number of connection weights. Applying neural network to a system with large number of inputs is complex as large ANN tends to get stuck in local minima [13]. In addition, most training algorithms require thousands of training epochs for small convergence errors because they start ANN training from initial random connection weights.

Our design is based on the PCA-CG training algorithm, initially introduced in [14]. This algorithm can train large-scale ANN models as it starts from non-random initial connection weights derived from the training data set. PCA-CG also uses Principal Component Analysis (PCA) to estimate the number of hidden neurons for training. However, due to processing limitations during validation in embedded system, we use a fixed number of 5 hidden neurons. The algorithm was successfully applied in a number of applications [15].

The capability of training large-scale neural networks eliminates the necessity of using only the most relevant input features. This fact is particularly valuable in case of flame detection, as the JTFA used in preprocessing stage produces a wide spectrum of time-frequency information, all of which is relevant for classification.

A. *Training Model*

The training model is based on the PCA-CG algorithm by Hugo Guterman[*] and Zvi Boger, which is described in detail in [14], [16], and [17]. The training algorithm consists of the following steps:

1) Form joint input-output data vector $X = x_p \cup y_p$, making N_p rows of matrix X represent the entire data set. The columns of X are scaled by subtracting the mean of each column from the values in it, and dividing the results by standard deviation of each column.

2) Calculate Σ_X [a x a] as

$$\Sigma_X = E\{(X - E\{X\})^T(X - E\{X\}\}$$

3) Determine the eigenvectors and eigenvalues of Σ_X. Select eigenvectors $\phi_1 \ldots \phi_r$ corresponding to the largest eigenvalues $\lambda_1 \ldots \lambda_r$ necessary for reconstructing X with a chosen information content ξ:

$$\mu_i = \frac{\lambda_i}{tr(\Sigma_x)} = \frac{\lambda_i}{\sum_{i=1}^{a} \lambda_i}$$

Then, assuming that λ_i and ϕ_i are ordered, the number of neurons in the hidden layer, ***r***, would be equal to the number of dimensions necessary to reconstruct the original information with a ζ degree of fidelity,

$$\sum_{\mathbf{i}=1}^{\mathbf{r}} \mu_{\mathbf{i}} \geq \xi$$

There are n inputs and m outputs, and $a = m+n$.

4) Compute the initial input to the hidden weights matrix W_H as follows (the last column are the bias values):

$$W_H = \begin{matrix} \phi_{11} & \ldots\ldots & \phi_{n1} & h_1 \\ \phi_{12} & \ldots\ldots & \phi_{n2} & h_2 \\ \ldots\ldots & \ldots\ldots & \ldots\ldots & \ldots\ldots \\ \phi_{1r} & \ldots\ldots & \phi_{nr} & h_r \end{matrix}$$

$$h_i = \sum_{j=n+1}^{a} \phi_{ij}^T \mathbf{E}\{\mathbf{X}_j\}$$

5) Compute the initial hidden-to-output weights matrix W_O (the last column are the bias values):

[*] Department of Electrical and Computer Engineering, Ben-Gurion University, Be'er Sheva, Israel

$$W_o = \begin{matrix} \phi_{(n+1)1} & \ldots & \phi_{(n+1)r} & u_{n+1} \\ \phi_{(n+2)1} & \ldots & \phi_{(n+2)r} & u_{n+2} \\ \ldots & \ldots & \ldots & \ldots \\ \phi_{a1} & \ldots & \phi_{ar} & u_a \end{matrix}$$

$$\mathbf{U}_{bias} = \sum_{i=r+1}^{a} \phi_i^T \mathbf{E}\{\mathbf{X}\}\phi_i = [u_1, u_2, \ldots, u_a]^T$$

Conjugate-gradient method similar to the one proposed in [20] is employed for searching the optimum weights.

This algorithm is different from other commonly used ANN training algorithms in the following characteristics:

1) It uses non-random initial connection weights, calculated from characteristics of training data.

2) The number of hidden neurons is kept small, usually 4 to 7.

3) Proprietary algorithms avoid and escape from local minima in the complex multi-dimensional error surface, encountered during the training.

4) For knowledge extraction, the Causal Index (CI), describing the magnitude and sign effect on any output when each input value is changed, is calculated from the trained ANN connection weights [18].

5) The behavior of the output of hidden neurons when the trained ANN is presented with data is used for error checking and clustering, by grouping similar patterns. Auto-associative ANN (AA-ANN) [19], in which the input vector is presented also as the output vector, can thus generate unsupervised clustering, important when no prior categorization is available.

B. Classification Model.

The ANN classification model implemented in the embedded system consists of 5 hidden and 1 output neuron, indicating either flame or non-flame condition. A unipolar neuron activation (sigmoid) function was used at the output of every neuron. The model of our implementation of the feed-forward network is depicted in Figure 2, and described in detail in the next section on experimental results.

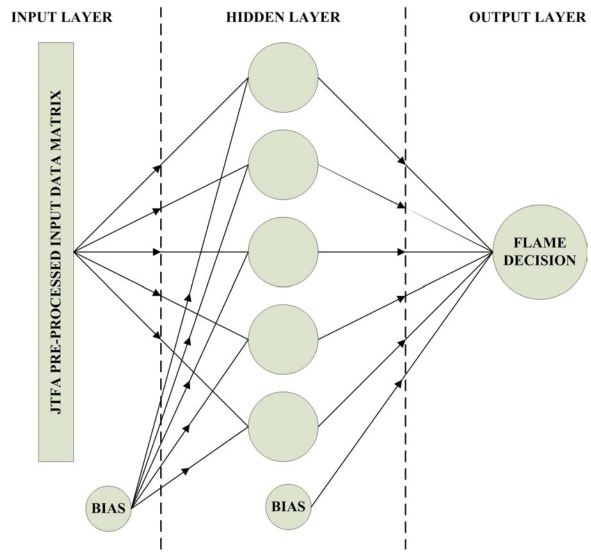

Figure 2. ANN Classification Model

IV. EXPERIMENTAL IMPLEMENTATION AND RESULTS

A. Signal Pre-Processing

The concept of ANN-based flame detection described in previous section has been implemented in the design of a General Monitors next generation IR flame detector [7]. It consists of four IR sensors responding to phenomena at different wavelengths of IR spectrum. Analog sensor signals were sampled at a rate of 10 milliseconds and converted into a digital format for signal preprocessing

As a part of JTFA, a 512-point Hamming data window followed by a 512-point FFT was applied to a signal segment of 5.12 seconds from each IR sensor. The first 256 points of FFT output contain non-symmetric frequency information in range 0 ~ 50 Hz with the resolution of 0.2 Hz. So, for four detectors, there were 1024 frequency inputs. In addition, we also averaged the raw signal information over the period of past 64 ms, which generated one more input point for each sensor. So the signal preprocessing stage produces 1028 total ANN input columns. Every 25 samples (250 ms), the FFT window were shifted in time, and data windowing with FFT calculation was repeated for the previous 512 samples. So, a new input feature sample for ANN was generated every 250 milliseconds.

Input signals were collected with the IR sensors observing various flame and non-flame conditions, including *n*-heptane, propane, and butane flames at distances of 0 to 250 feet as well as direct and reflected sunlight, arc welding, random hand wave, modulated heat, flashlight, and other non-flame sources at various distances.

B. ANN Training and Testing

The training was conducted on a data set collected from the IR sensors observing relevant environmental conditions. The training program ran in MATLAB 7.0 software on a Sun Blade 2500 workstation. A total of 27000 input samples of 1028 (matrix of 27000x1028) were used, out of which 70% of the data was used for training set, and 30% - for an independent testing set. There was a single target column indicating either flame (1) or non-flame (0) condition.

In several trials, the training Root-Mean Square (RMS) error rate was below 5%. Depending on the variety of input environmental conditions and ordering of input data, the training algorithm converged in 150-300 training epochs. RMS results for training and testing are presented in Figure 3.

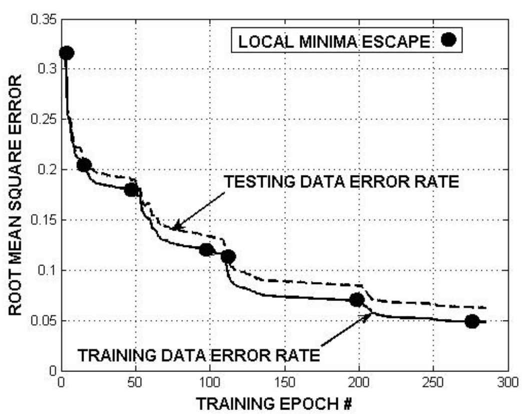

Figure 3. ANN Training RMS Error Output

Using the connection weights obtained from training (loaded as constants), the ANN classification scheme with 5 hidden and 1 output neurons, and a unipolar activation function, was implemented in virtual floating-point arithmetic on Texas Instruments F2812 Digital Signal Processor (DSP).

C. ANN Output Post-Processing

The ANN output value on 0 to 1 scale is generated every 250 milliseconds. The post-processing scheme uses two variable parameters: *flame threshold* and *sensitivity delay* as shown below in Figure 4.

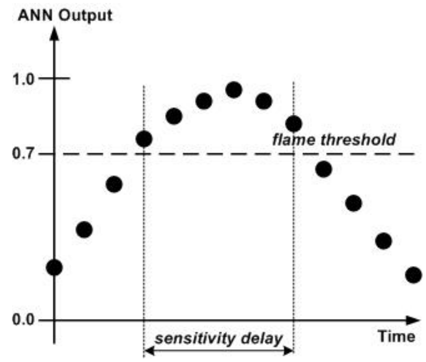

Figure 4. ANN Output Post-Processing Scheme

The flame threshold value indicates a limit above which an ANN output can be considered as a flame condition. The sensitivity delay sets the number of ANN outputs necessary to make a confident decision about the presence of flame detection.

In our implementation, the flame threshold value was set to 0.7, and sensitivity delay was set to 18 ANN outputs at 250 milliseconds, or 4.5 seconds total. Both values were derived experimentally. The latter value is important from engineering point of view because it also defines the response time of the detection system to a flame condition.

V. Conclusion

A design for an industrial IR flame detection using ANN is presented. JTFA using STFT was applied to identify relevant signal frequencies as input features for the ANN. Other, more advanced methods, such as DWT can be applied to obtain a better set of input features and subsequently better training and classification.

The ANN training was based on the CG method along with PCA used for initializing the connection weights with non-random values. It was trained on real environmental data. A classification scheme based on 5 hidden and 1 output neurons was implemented on the DSP. Higher memory and processing capabilities could potentially enable better training and classification. Using more output neurons in such case would enable classification not only by flame/non-flame scenario, but also to identify the types of flame and non-flame sources by analyzing input IR signals.

The design described in this paper has been implemented in an industrial IR flame detector [7] by General Monitors, Inc (GMI). The flame detector using the ANN classifier provides for longer-range (up to 250 ft) flame detection than that currently provided by the expert systems. At the same time, it also provides for exceptional discrimination against non-flame sources of radiation.

References

[1] Shankar Baliga, Herb Rabe, Brett Bleacher. "Digital Multi-Frequency Flame Detector," *U.S. Patent No. 6, 150, 659*, November 21, 2000.

[2] Fred Schuler. "Dual Wavelength Fire Detection Method and Apparatus," *U.S. Patent No. 5, 850, 182*, December 15, 1998.

[3] Ephraim Goldenberg, Tal Olami, Jacob Arian. "Method For Detecting A Fire Condition", *U.S. Patent No. 5, 373, 159*, December 13, 1994.

[4] J.A. Milke and T.J. McAvoy. "Analysis of Signature Patterns for Discriminating Fire Detection with Multiple Sensors," *Fire Technology*, Second Quarter 1995.

[5] Y. Okayama. "Approach to Detection of Fire in Their Very Early Stage by Odor Sensors and Neural Net," *Proceedings of Third International Symposium of Fire Safety Science*, pp. 955-964, 1991.

[6] Yonggang Chen, Michael A. Serio and Sandeep Sathyamoorthy. "Development of a Fire Detection System Using FT-IR Spectroscopy and Artificial Neural Networks," *Proceedings of the Sixth International Symposium of the International Association for Fire Safety Science*, pp. 791-802, 1999.

[7] Gary Shubinsky, Shankar Baliga, Javid Huseynov, Zvi Boger. "Flame Detection System," *U.S. Patent No. 10 / 894,570*, submitted in July 2004.

[8] Tomasz P. Zielinski. "Joint time-frequency resolution of signal analysis using Gabor transform." In: *IEEE Transactions on Instrumentation and Measurement*, Volume 50, No.5, October 2001

[9] David Swanson. *Signal Processing For Intelligent Sensor Systems*, Marcel Dekker, Inc., 2000.

[10] Dennis Gabor. *Theory of communication*, J. IEE (London), vol.93, pp. 429-457, 1946.

[11] Metin Akay. *Time Frequency and Wavelets in Biomedical Signal Processing*, Wiley-IEEE Press, 1997.

[12] E. M. Johansson, F. U. Dowla, D. M. Goodman. "Backpropagation learning for multi-layer feed-forward neural networks using the conjugate gradient method," *International Journal of Neural Systems*, 2(4), pp. 291-301, 1992.

[13] Zvi Boger. "Artificial Neural Networks Methods for the Identification of the Most Relevant Genes from Gene Expression Array Data," *International Joint Conference on Neural Networks*, ICJNN'03, Portland, OR, July 2003.

[14] Hugo Guterman. "Application of Principal Component Analysis to the design of neural networks," *Neural, Parallel and Scientific Computations*, vol. 2, pp. 43-54, 1994.

[15] Zvi Boger. "Who is afraid of the BIG bad ANN?" *Proceedings of the International Joint Conference on Neural Networks, IJCNN'02, 1215*, pp. 2000-2005, Honolulu, Hawaii, 2002.

[16] Zvi Boger and Hugo Guterman. "Knowledge extraction from artificial neural networks models," *Proceedings of the IEEE International Conference on Systems Man and Cybernetics, SMC'97,* Orlando, Florida, pp. 3030-3035, 1997.

[17] Zvi Boger. "Selection of the quasi-optimal inputs in chemometric modeling by artificial neural network analysis," *Analytica Chimica Acta*, vol. 490 (1-2), pp. 31-40, 2003.

[18] K. Baba, I. Enbutu, and M. Yoda. "Explicit representation of knowledge acquired from plant historical data using neural network," *Proceedings of the International Joint Conference on Neural Networks*, vol. 3, pp. 155-160, 1990.

[19] Zvi Boger. "Finding patient cluster's attributes by auto-associative ANN modeling," *Proceedings of the International Joint Conference on Neural Networks*, *IJCNN'03*, Portland, OR, pp. 2643-2648, July 2003.

[20] J. Leonard and M. A. Kramer. "Improvement of the Backpropagation Algorithm for training neural networks", *Computers Chemical Engineering*, 14(3), pp. 337-341, 1990.