FACTA UNIVERSITATIS
Series: **Electronics and Energetics** Vol. 26, N^o 3, December 2013, pp. 215 - 225
DOI: 10.2298/FUEE1303215M

IMPLEMENTING TEMPLATE MATCHING LOGIC IN A STANDARD FLASH MEMORY CELL

Mitar Milacic, Sima Dimitrijev

Queensland Micro- and Nanotechnology Centre and Griffith School of Engineering,
Griffith University, Nathan, Qld. 4111, Australia

Abstract. *Current research into classification methods is almost exclusively software based, resulting in systems that perform well but are invariably slow when faced with large databases. The goal is therefore to create a hardware classification system that is much faster. In this paper, we introduce the concept of template matching logic and propose the use of a standard flash memory cell array to perform bit by bit template matching. The proposed system is based on a novel architecture that is unique and separate from existing architectures that make use of flash memory cell arrays. Verification is achieved by speech recognition simulations on the TIMIT database. Simulations of the system show results of 94.5 % recognition accuracy on clean words and 88.0 % recognition accuracy on test words with a signal-to-noise ratio of 5 dB. The results compare favorably to similar isolated word recognition tasks performed with software based methods.*

Key words: *Hardware template matching, classification, flash memory cell*

1. INTRODUCTION

Data classification problems arise in many forms: speech and face recognition, pattern recognition, medical diagnoses, etc. Software based approaches to these problems can obtain good accuracies, however, they are invariably slow-constrained by the speed of the operating systems that they run on. Hardware based classification systems remove these constraints; they are often able to obtain the same results as the software based systems at a much faster speed. An example of a hardware based classification system is a hardware implementation of artificial neural networks (ANNs). Hardware implementations of ANNs combine memory storage and data processing in a non-Von Neumann architecture.

The high number of connections and large cell sizes required by ANNs severely limit the size of the network that can be built in hardware and by extension also limit the functional complexity of the ANN. The limited functional complexity can be illustrated by examples of analogue implementation of ANNs, which have been the most significant approach to developing hardware-based intelligent systems [1]-[8]. Horio *et al* [8] have demonstrated a neuron chip containing only five neurons. Dede and Sazli [9] have used a

Received November 28, 2013
Corresponding author: Sima Dimitrijev
Queensland Micro- and Nanotechnology Centre and Griffith School of Engineering, Griffith University, Nathan, Qld. 4111, Australia
(e-mail: s.dimitrijev@griffith.edu.au)

multilayer perceptron ANN to create a speech recognition system, using 50 neurons in just the two hidden layers. Given the current technology available, a hardware implementation of an ANN-based speech recognition is not a realistic option.

Template matching is the most fundamental and generic classification technique. Recent research into template matching has shown good results for face recognition [10]. It has been shown that template matching is very robust and applicable to different problems that arise from natural variability, such as the presence of occlusions in the case of face recognition and noise in the case of speech recognition. It is this ability to handle unpredictable changes in the environment that makes template matching so useful for real-world applications. Further, template matching closely matches the use of the synaptic memory of the brain.

Speech recognition presents a difficult problem. Current research is focused on hidden Markov models (HMMs) and ANNs [9], [11]-[13]. Both the HMM and ANN approaches to speech recognition are complex software-based systems that are slow and require a lot of processing power. Han *et al* [14], [15] have demonstrated that a software-based speech recognition system can be implemented in hardware. However this approach is not ideal in terms of maximized performance of the utilized hardware. A more effective approach is to design the system from basic hardware principles.

In this paper, we propose a new concept of template matching logic (TML) where a single input bit is compared to a memorized template bit to produce a single output bit. We also propose the implementation of TML using a standard flash memory cell and create a hardware template matching system using a standard flash-memory cell array. The proposed hardware template matching system is verified on a speech recognition task by computer simulations of a word-recognition problem on the standard TIMIT database. Speech recognition was chosen to evaluate the proposed system because it is known as a difficult task that requires complex computations when the standard speech-recognition approaches are applied.

2. PROPOSED SYSTEM DESIGN

Template matching is the most fundamental classification technique, and has been proved successful in software implementations. However, template matching requires a large number of memorized templates to perform well on difficult tasks, which can considerably slow down the recognition process. To address the problem of speed, to enable portable applications, and to enable improvement in the recognition rates by increasing the number of memorized templates, we propose a hardware implementation of the template matching method.

Flash-memory technology is the dominant semiconductor-based technology for nonvolatile memories and is used in a variety of consumer products, such as mobile phones, photo cameras, video cameras, music players, etc. In all these applications, the flash-memory technology is used in either NAND or NOR configuration of memory arrays. The hardware template matching system, proposed in this paper, utilizes the standard flash memory cell as found in the NOR memory arrays but its use is novel and different from both NAND and NOR configurations. The fundamental difference is in the proposed use of the standard flash-memory cell as both a processing and a memory unit, which departs from the Von Neumann architecture that limits the role of memory units to data storage.

Using the principles shown in this paper, as well as standard circuit design, a trained engineer will be able to design a complete system for any pattern recognition problem.

2.1. Template matching logic

The core of the hardware implementation of template matching is the use of a single memory cell, such as the cell in a flash memory cell array, at the level of a single bit-we refer to this as template matching logic (TML). TML is a new type of architecture, one that is not related to either NAND or NOR architectures. This section describes in detail the flash implementation of TML.

Table 1 shows that conceptually TML is similar to a logical AND operation, however, TML has only a single input, while the second bit used for comparison is memorized in the TML unit.

Table 1 The truth table for a single bit TML

Input	Memorized template bit	Output
1	1	1
1	0	0
0	0	0
0	1	0

Table 2 shows how the transfer characteristic of a standard flash memory cell (both erased and programmed states are shown in Fig. 1) can be utilized to implement TML. Initially the flash cell is in the erased state, so when the control-gate voltage (V_{CG}) is set high (V_H) the current through the bit line (I_{BIT}) is high (I_H), which corresponds to an output of **1**. If the cell is in its programmed state and V_{CG} is set at V_H, the current I_{BIT} is low (I_L), which corresponds to an output of **0**. If V_{CG} is set low (V_L), the current I_{BIT} is low (I_L) and the output remains **0**, irrespective of the cell state.

Table 2 Flash-cell implementation of TML

Input (V_{CG})	Memorized bit (cell state)	Output (I_{BIT})
1 V_H	1 Erased	1 I_H
1 V_H	0 Programmed	0 I_L
0 V_L	0 Programmed	0 I_L
0 V_L	1 Erased	0 I_L

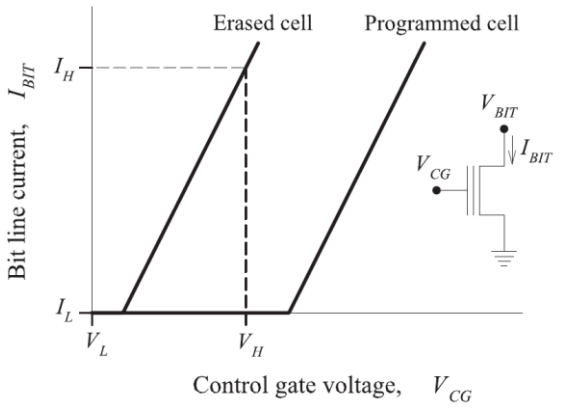

Fig. 1 Transfer characteristics of a cell in a standard flash memory cell array

2.2. Template similarity score

Each classification task will use templates of different sizes, depending on the task and the feature extraction method used. The example in Table 3 is with template vectors of length equal to four elements. Applying element-by-element TML, the elements in the vector of the input template (P) are compared to the elements in the vector of the memorized template (T), resulting in the output vector O. Mathematically, each element in vector O is given by the multiplication of the corresponding elements in P and T ($O_i = P_i * T_i : i = 1, 2, 3, 4$). The similarity score, S_O, is obtained by adding up the elements in O ($S_O = 2$ in the example shown in Table 3).

Table 3 TML applied to specific example

Input template	Memorized template	Output array
$P = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}$	$T = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$	$O = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$

Similarity score (S_O)

$$S_o = \sum_{i=1}^{4} P_i * T_i = \sum_{i=1}^{4} O_i = 2$$

In hardware implementation, four flash cells are connected together into a flash-cell vector, as shown in Fig. 2. The template vector, T, is memorized in the flash-cell vector using the standard process for storing data. The voltage V_{BIT} is applied to allow the cells in the vector to be turned on. A **0** element in P_i sets the voltage $V_{CG\text{-}i}$ to V_L, whereas a **1** element in P_i sets $V_{CG\text{-}i}$ to V_H. If the cell is in the erased state (**1**) and V_H is applied to the control gate, then the current $I_{BIT\text{-}i} = I_H$ will flow to ground through the source line. Taking flash cell C_2 as an example, the current $I_{BIT\text{-}2}$ is given by the following equation:

$$I_{BIT\text{-}2} = I_H * T_2 * P_2 = I_H \tag{1}$$

In the flash cell vector, it is only possible to measure the total current in the bit line ($I_{BIT\Sigma}$); it is not possible to measure the current running through a single flash cell. However, $I_{BIT\Sigma}$ is the summation of all the currents flowing through all the flash cells connected to that particular bit line. Mathematically $I_{BIT\Sigma}$ can be determined by the following equation:

$$I_{BIT\Sigma} = I_H \cdot \sum_{i=1}^{N} T_i \cdot P_i \tag{2}$$

where N is the number of elements in the flash-cell vector. Plugging in the values for templates T and P, for the example shown in Table 3, into Eq. 2 gives $I_{BIT\Sigma} = 2I_H$. The similarity score can then be obtained as the normalized current $I_{BIT\Sigma}$:

$$S = \frac{I_{BIT\Sigma}}{I_H} \tag{3}$$

Because $I_{BIT\Sigma} = 2I_H$ in the considered example, the similarity score is $S = 2$, which corresponds to the value in Table 3.

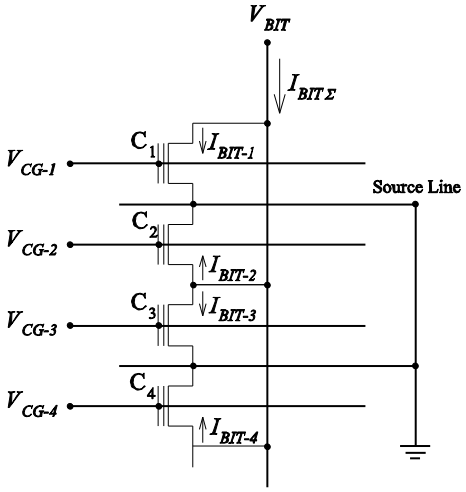

Fig. 2 Flash-cell vector

2.3. Multiple template comparison

The flash memory cell array shown in Fig. 3 is used to compare the input template to multiple memorized templates simultaneously. The system performs at its peak with the templates each memorized on a separate bit lines, allowing for the similarity scores to be calculated in parallel. If the employed feature extraction method produces a multiple dimensional template, the template is converted into a 1D vector by stacking the columns. That way, the input template can effectively be compared to all of the memorized templates in parallel, which represents a massive speed up over software based template-matching methods. The simultaneous comparison to many memorized templates will require a winner-take-all circuit to compare the output currents of the bit lines ($I_{BIT\Sigma-1}$ to $I_{BIT\Sigma-M}$, where M is the number of memorized templates) and to uniquely identify the memorized template that resulted in the highest current. The identity of the bit line with the highest current is passed on to either a computer or a microchip for further processing, depending on the application.

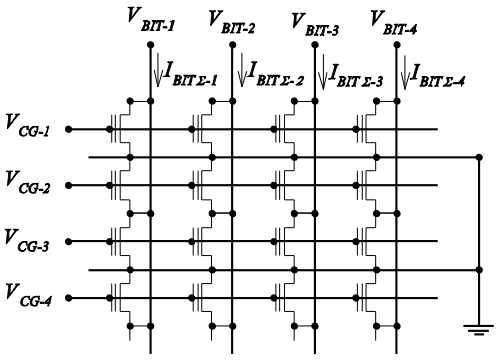

Fig. 3 Flash-cell array

2.4. Template digitization

The majority of feature extraction methods produce templates with analog values. To enable the application of the binary version of TML, the analog templates have to be encoded into a binary format via the digitization process. Additionally, digitization ensures that each of the flash memory cells are either completely on ("1") or completely off ("0"), thereby circumventing problems that would otherwise arise due to process variations.

A simple digitization process is shown graphically in Fig. 4. To explain this process, consider a single analog value, x, which is normalized to be within the range 0-1. The digitized binary vector that corresponds to this value will have B bits ($B = 5$ in Fig. 4). The elements of the digitized template are determined by the following equation:

$$X_k = \begin{cases} 1, & x > thres(k) \\ 0, & otherwise \end{cases} \tag{4}$$

where X is the digitized matrix, k is the index denoting the coding bit ($k = 1, \ldots, B$), and *thres* is a vector containing the thresholds for each bit level:

$$thres(k) = \frac{k}{B+1} \text{ for } k = 1, 2, ..., B \tag{5}$$

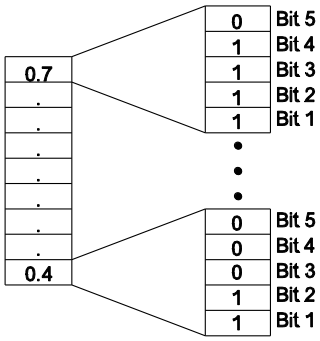

Fig. 4 Illustration of the digitization process converting each analogue value into a binary vector of length 5. The threshold for each bit in the vector is given by Eq. 5 ($B = 5$)

When it is not practical to limit the range of the templates between 0 and 1, the thresholds used for each bit level can be increased by multiplying the *thres* vector with an appropriate scalar.

The binary vectors resulting from Eq. 4 are stacked to create a 1D binary vector, as illustrated in Fig. 4. This binary vector can then either be memorized into the flash cell array or used as an input to the flash cell array and compared to the memorized template.

If the vector is larger than what can be placed on a single array, then the vectors can be split up over several flash cell arrays and the output currents of the appropriate bit lines added up.

3. VERIFICATION OF THE PROPOSED SYSTEM BY COMPUTER SIMULATION

Speech recognition is known as a difficult task, particularly when noise corrupts the input signal. We selected a word recognition task under noisy conditions to verify the proposed hardware implementation of template matching system by computer simulation.

3.1. Database used for speech recognition

The commonly used TIMIT database [16] was used for the verification. The TIMIT database is recorded in American English with 620 male and female speakers, speaking in eight different dialects. An analysis of the sentences generally used for speech recognition in the TIMIT database revealed that while they contained over 7,000 unique words, very few of the words repeated often enough to be of use for a template matching recognition task. Of those that did repeat often enough, most were very short, for example: *a*, *in*, *the*, etc. Using these words would be more indicative of phoneme recognition than word recognition due to their short duration. To avoid this problem the word recognition task was performed using *sa1* and *sa2* sentences that are usually reserved for speaker recognition studies. These sentences are:

- She had your dark suit in greasy wash water all year.
- Don't ask me to carry an oily rag like that.

Outside of a laboratory it is very unlikely to find a noise free environment, therefore to provide a more realistic scenario the system is tested with varying amounts of white Gaussian noise added. Additionally, the introduction of noise to the speech signal makes the recognition problem harder and allows for a better comparison of different classifiers. The white Gaussian noise signal was created by an algorithm that generates pseudo-random numbers according to the Gaussian distribution with the mean of zero and the standard deviation of one. The amplitude of the generated noise signal was then set so that the desired signal-to-noise ratio (SNR) was achieved on a sentence by sentence basis.

3.2. Feature extraction

The features were extracted using standard signal processing techniques to convert the input waveform into a Bark-scale spectrogram. The waveform was segmented into overlapping frames (25 ms in length with a 10 ms shift) and a Hamming window applied for analysis [17]. A discrete Fourier transform was applied to each frame, using 512 frequency bins, and the frequency spectrum was re-scaled using a *Bark* scale filter bank [17]. This transform converts the speech waveform into 21 frequency "channels". Mean-variance normalization was applied on the entire sentence using a moving-window filter of size [1 x 200 (2*s*)] [18]. The utterances were segmented into words using data provided in the TIMIT database. Zeros were inserted after each word to ensure every template was 100 frames in length (1 second of speech). The 21 x 100 sized templates were digitized using the process described in section 2.4. The speech templates were digitized using 5, 10 and 20 bits.

Unique to speech recognition is that the length of the word can also be used for classification. The method that was used to enable comparisons of the word lengths is illustrated in Fig. 5. This method also enables the necessary normalization of the similarity score with respect to the word length. In this method, the match of **1** binary values in the input and memorized templates is added to a match of **0** binary values. This ensures that

the similarity score of a short word with a small number of matched **1** values is not confused with a poor match of long words. To implement the additional matching of **0** values, both the input and the memorized templates are inverted. The inverted memorized template is stored in a flash-cell array, labeled by *T′* in Fig. 5. When the inverted input template (*P′* in Fig. 5) is compared to the inverted memorized template, output **1** values correspond to matched inverted **0** values in the input and memorized templates.

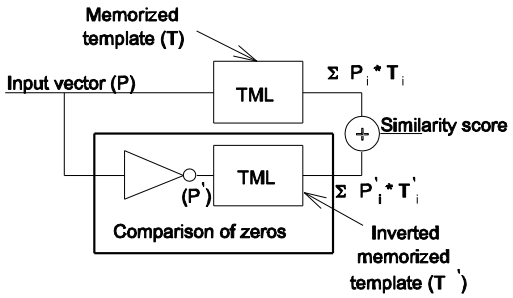

Fig. 5 Illustration of the steps required to normalize the similarity score according to word length. By inverting the input and the memorized template, TML is used to count the number of matching zeros

To reflect the addition of the inverted templates, Eq. 2 is changed as follows:

$$I_{BIT\Sigma} = I_H \Sigma_{k=1}^{N} [P_k * T_k + P'_k * T'_k] \tag{6}$$

where *T′* and *P′* are the inverted versions of templates *T* and *P*, respectively, and *N* is the number of binary elements in the templates.

For each of the 21 unique words 468 templates are memorized for a total of 9,828 memorized templates, with an additional 3,402 templates (for each SNR level) used as the input to test the proposed system.

A hardware implementation of this speech recognition system would require that the feature extraction be implemented in software on either a computer or a microchip. This would in no way change the template-matching system proposed in this paper.

3.3. Results and discussion

Figure 6 shows that the proposed method performs well on a word recognition problem across various SNRs. At low SNRs (5 dB) the number of bits used for digitization has a greater impact on the recognition accuracy. This impact is diminished for the clean results. A side effect of increasing the number of bits for digitization is an increased number of memory cells required to store each template, therefore the number of bits used in digitization needs to be optimized based on the requirements of each application. Most of the accuracies in Fig. 6 are above 90 %. Frikha and Hamida [19] used a different classification method on a similar word recognition task on the TIMIT database (using just ten of the 21 words as classes for recognition). They reported a recognition accuracy of 98.99 % with a clean input and recognition accuracies of 77.13 %, 46.75 % and 34.04 % for test inputs with white Gaussian noise at SNRs of 20dB, 10dB and 5dB, respectively. While the clean results presented by Frikha and Hamida are higher than the

results of the method proposed here, all of the noisy results are significantly lower than the results of the proposed method. The results shown in Fig. 6 demonstrate that the proposed method is robust to noise and would perform well in most real-life situations.

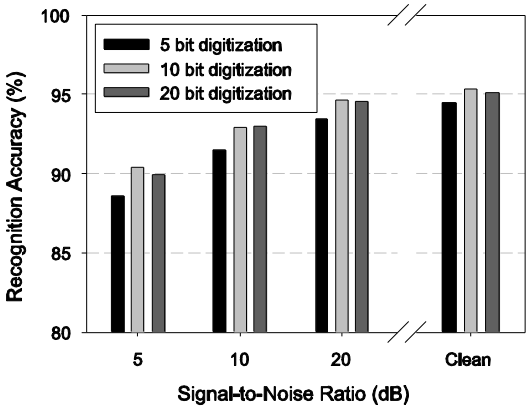

Fig. 6 Recognition accuracies obtained by the TML approach with the Bark-scale spectrogram as the features

To further compare the performance of the proposed hardware template matching system with that of software based classifiers, experiments were performed using dynamic time warping (DTW)[1]. Table 4 shows the recognition results using these two software based classifiers. On a clean signal DTW performs better however a significant drop in recognition is seen with a decrease in SNR. The padded similarity method shows results that are similar to those of the proposed system.

Table 4 Recognition results for isolated word recognition using software based classifiers

SNR (dB)	DTW
Clean	97.08
20	96.26
10	90.87
5	82.34

The proposed template matching method is generic and is therefore capable of performing well with other feature extraction methods. The standard set of features used in speech recognition are Mel-frequency cepstrum coefficients (MFCCs) [21]. Although MFCCs are not designed to work with a template matching method, we tested the application of these features in the proposed system with a 10 bit digitization. The following recognition accuracies were obtained: 90.6 %, 85.4 %, 66.4 % and 49.8 % on clean, 20 dB, 10 dB, and 5 dB input signals, respectively. The wide range of features that can be used with TML allows for TML to be used in many different applications.

[1] The DTW method used is from ref. [20] and the MATLAB code is available for download from http://labrosa.ee.columbia.edu/matlab/dtw/.

4. System-Application Discussion

The proposed hardware template matching system has many advantages over a software-based systems. The most important advantage is the speed of classification. Tasks that require several minutes or hours to complete using a software based system can be accomplished by the proposed system in fractions of a second, which means that these tasks can be performed in real time. The system allows for large databases to be processed at a speed that cannot be matched with a software based system. Another advantage is the reduced cost associated with building the proposed system as it is built on existing technology and does not require state of the art CPUs to perform well. A cheap, malleable, and fast speech recognition system can be coupled with a mobile phone to create user specific commands for mobile banking or other mobile services even if the user is not a native English speaker. The ability of the system to be programmed with new templates allows for new users to be added or old users removed. Additionally the system can be coupled with everyday objects to give them the ability to understand speech, either in single-user or multiple-user applications.

The proposed template matching system is not limited to speech recognition. Almost any type of template can be used in this system to create a fast and accurate classification system. The system can be used for face recognition applications to identify people in very large databases, very quickly. The inclusion of a threshold on the similarity score can create a verification system that can be used for security applications. It is also possible to perform medical diagnoses based on templates. X-rays can be compared either to past X-rays from the same patient or other patients in order to diagnose a problem. 3D MRI data can also be processed quickly and the similarity score can quantify the changes. The templates can also be finger prints or DNA, both of which require large databases for a good match.

The intrinsic speed of the proposed system can be utilized in many different ways. All these possible applications provide avenues to move computing technology beyond Moore's law and to reduce its reliance on slow software based systems.

5. Conclusion

In this paper, we propose a new concept called template-matching logic, and a method to implement it in a standard flash memory cell array. The proposed implementation of template matching in hardware enables complex classification tasks to be performed much faster in comparison to the existing software-based systems. The hardware system outlined in this paper can be designed by trained engineers in order to utilize the proposed method in a specific application. The proposed method was applied to a word recognition task to enable simulation-based verification on a standard speech-recognition database. The results demonstrate robust performance of the proposed system with different features and under various signal-to-noise ratios.

REFERENCES

[1] M. Valle, "Analog vlsi implementation of artificial neural networks with supervised on-chip learning", *Analog Integr. Circuits Signal Process.*, vol. 33, pp. 263–287, December 2002.

[2] G. Ascia, V. Catania, and M. Russo, "Vlsi hardware architecture for complex fuzzy systems", *IEEE Transactions on Fuzzy Systems*, vol. 7, pp. 553–570, October 1999.

[3] R. Mason, W. Robertson, and D. Pincock, "An hierarchical vlsi neural network architecture", *IEEE Journal of Solid-State Circuits*, vol. 27, pp. 106–108, January 1992.

[4] L. O. Chua, Cellular Neural Networks and Visual Computing: Foundations and Applications. Cambridge University Press, Cambridge, 2002.

[5] B. Linares-Barranco, A. Andreou, G. Indiveri, and T. Shibata, "Guest editorial – special issue on neural networks hardware implementations", *IEEE Transactions on Neural Networks*, vol. 14, pp. 976–979, September 2003.

[6] M. Milev and M. Hristov, "Analog implementation of ann with inherent quadric nonlinearity of the synapses", *IEEE Transactions on Neural Networks*, vol. 14, pp. 1187–1200, September 2003.

[7] A. Gopalan and A. Titus, "A new wide range euclidean distance circuit for neural network hardware implementations", *IEEE Transactions on Neural Networks*, vol. 14, pp. 1176–1186, September 2003.

[8] Y. Horio, K. Aihara, and O. Yamamoto, "Neuron-synapse ic chip-set for large-scale chaotic neural networks", *IEEE Transactions on Neural Networks*, vol. 14, pp. 1393–1404, September 2003.

[9] G. Dede and M. H. Sazl, "Speech recognition with artificial neural networks", *Digital Signal Processing*, vol. 20, no. 3 pp. 763–76, 2010.

[10] A. P. James and S. Dimitrijev, "Face recognition using local binary decisions", *IEEE Signal Processing Letters*, vol. 15, pp. 821–824, 2008.

[11] S. Scanzio, S. Cumani, R. Gemello, F. Mana, and P. Laface, "Parallel implementation of artificial neural network training", in *2010 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pp. 4902–4905, March 2010.

[12] K.-F. Lee and H.-W. Hon, "Speaker-independent phone recognition using hidmen markov models", *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, pp. 1641–1648, November 1989.

[13] S. Kruger, M. Schaffoner, M. Katz, E. Andelic, and A. Wendmuth, "Mixture of support vector machines for hmm based speech recognition", in *18th International Conference on Pattern Recognition, 2006, ICRP 2006*, vol. 4, pp. 326–329, 2006.

[14] W. Han, K.-W. Hon, C.-F. Chan, T. Lee, C.-S. Choy, K.-P. Pun, and P. Ching, "An hmm-based speech recognition ic", in *Proceedings of the 2003 International Symposium on Circuits and Systems, 2003, ISCAS 03*, vol. 2, pp. II–744–II–747, May 2003.

[15] W. Han, K.-W. Hon, C.-F. Chan, T. Lee, C.-S. Choy, K.-P. Pun, and P. Ching, "A real-time chinese speech recognition ic with double mixtures", in *Proc. 5th International Conference on ASIC, 2003*, vol. 2, pp. 926–929, October 2003.

[16] W. M. Fischer, G. R. Doddington, and K. M. Goudie-Marshall, "The darpa speech recognition research database: specifications and status", in *DARPA workshop on speech recognition*, pp. 93–99, 1986.

[17] J. W. Picone, "Signal modeling techniques in speech recognition", in *Proceedings of the IEEE*, pp. 1215–1247, 1993.

[18] O. Viikki and K. Laurila, "Cepstral domain segmental feature vector normalization for noise robust speech recognition", *Speech Communication*, vol. 25, no. 13, pp. 133-147, 1998.

[19] M. Frikha and A. B. Hamida, "Noise robust isolated word recognition using speech feature enhancement techniques", *Journal of Applied Sciences*, pp. 3935-3942, 2007.

[20] R. Turetsky and D. Ellis, "Ground-truth transcriptions of real music from force-aligned midi synthesis", in *4th International Symposium on Music Information Retrieval*, pp. 135-141, October 2003.

[21] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences", *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 28, pp. 357-366, August 1980.