A SPEECH EMOTION RECOGNITION FRAMEWORK BASED ON LATENT DIRICHLET ALLOCATION: ALGORITHM AND FPGA IMPLEMENTATION

Mohit Shah, Lifeng Miao, Chaitali Chakrabarti, and Andreas Spanias

School of Electrical, Computer and Energy Engineering
Arizona State University, Tempe, AZ 85287, USA

ABSTRACT

In this paper, we present a speech-based emotion recognition framework based on a latent Dirichlet allocation model. This method assumes that incoming speech frames are conditionally independent and exchangeable. While this leads to a loss of temporal structure, it is able to capture significant statistical information between frames. In contrast, a hidden Markov model-based approach captures the temporal structure in speech. Using the German emotional speech database EMO-DB for evaluation, we achieve an average classification accuracy of 80.7% compared to 73% for hidden Markov models. This improvement is achieved at the cost of a slight increase in computational complexity. We map the proposed algorithm onto an FPGA platform and show that emotions in a speech utterance of duration 1.5s can be identified in 1.8ms, while utilizing 70% of the resources. This further demonstrates the suitability of our approach for real-time applications on hand-held devices.

Index Terms— emotion recognition, affective computing, latent Dirichlet allocation, FPGA implementation

1. INTRODUCTION

Low-power hand-held devices today are capable of supporting high-end applications related to speech or speaker recognition or video-based activity recognition. Looking forward, we envision such devices to be capable of recognizing a user's emotional state and to play a significant role towards improving human-machine interaction. Speech has proven to be a good indicator of emotional content [1], with applications in diverse areas such as medical analysis, security and surveillance, personal memory aids and lifelogs [2, 3]. For successful speech-based emotion recognition in hand-held devices, the challenge lies in the design of algorithms to represent emotions with high accuracy and low implementation complexity.

Experiments in speech-based emotion analysis have identified a 3-D space, comprising of activation, valence and dominance, to provide a good representation of the underlying emotional content [4]. Activation indicates the arousal level in speech. For example, anger shows higher activation compared to sadness. Valence attributes to an emotion's positive or negative nature. Thus, happiness has a positive valence compared to disgust. Dominance indicates the strength of the perceived emotion. For instance, fear is a stronger emotion compared to boredom, thus, more dominant. The primary goal in emotion recognition is the identification of emotions either as discrete categories, such as happiness vs. anger, or a continuous representation of the emotional state in this 3-D space.

This work was supported in part by a grant from NSF CSR 0910699.

A typical speech-based emotion recognition system is shown in Figure 1. Raw speech is indexed by low-dimensional vectors, commonly known as features, followed by modeling via well-known methods such as hidden Markov models (HMMs) [5] or Gaussian mixture vector auto regressive models (GMVARs) [6]. These methods have shown to characterize the temporal evolution of features well and demonstrate a performance suitable for the task of emotion recognition. However, such temporal models are prone to severe degradation in real-world scenarios, where the probability of interruption by ambient sounds or multiple speakers is high. These limitations can be overcome by enforcing higher order temporal dependencies, however, at the cost of increased computational complexity.

In this paper, we address this issue by modeling the emotional content via latent Dirichlet allocation (LDA) [7]. LDA is an unsupervised, hierarchical Bayesian model for discrete data, originally intended for document representation. A corpus of emotional speech can be modeled via LDA. An utterance represents a document and its derived features represent the discrete data (observations). Each document is modeled as a mixture over multiple hidden topics; each topic, in turn, is a mixture over discrete observations. Contrary to the conventional approach in HMMs, LDA simplifies the model by ignoring temporal structure, and yet it captures significant intra-document statistical structure.

Our contributions mainly include the adaptation of LDA coupled with support vector machines (SVMs) for a discrete categorization of emotions. Our proposed approach clearly outperforms HMM or GMVAR-based methods in terms of recognition accuracy. Moreover, the simplicity of its implementation and amenability to parallelization makes it highly appealing for real-time applications. This claim is validated by an FPGA-based implementation of the complete framework.

2. BACKGROUND

The process of emotion recognition from speech is illustrated in Figure 1. Prosodic features including pitch or energy contours, voice quality features such as speaking rate or harshness and spectral features such as Mel frequency cepstral coefficients (MFCCs) have all been identified to suitably characterize the emotional content [8]. We have chosen energy and MFCCs as features because of their widespread use in speech recognition, thus allowing for better integration with existing systems. A commonly used feature extraction procedure is described as follows. Raw speech is high-pass filtered with a pre-emphasis coefficient of 0.97. Hamming windows of duration 25 ms are used to extract features at a rate of one feature vector every 10 ms. The features include log energy and the first 12 MFCCs and their corresponding first and second derivatives, resulting in a 39-dimensional vector. The computational complexity of feature extraction for 25ms of speech is given in Table 1, where operations de-

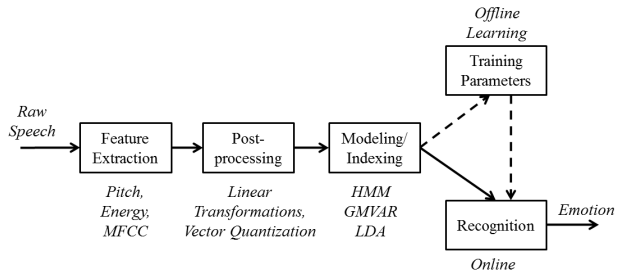

Fig. 1. Block diagram of a speech emotion recognition system.

Table 1. Computational complexity of feature extraction.

Feature extraction	Operations
Pre-emphasis	400
Hamming window	400
Log energy	400
MFCC	19424
Total	20624

note the number of multiplications. Generally, feature extraction is followed by a post-processing step. This involves principal component analysis (PCA) or vector quantization (VQ) routines for further dimensionality reduction.

The distribution and variation of such features is modeled using HMMs, GMVAR or LDA as proposed here. The choice of classifier is mainly dependent on the application requirements. For instance, support vector regression (SVR) is appropriate if a continuous representation of the user's emotional state is desired, while SVMs or artificial neural networks (ANNs) are more suitable for a discrete categorization of emotions. This overall process of learning the model and classifier parameters is usually performed offline over a fixed corpus of utterances. Recognition is described as the classification or representation of emotions in unknown utterances using the learnt parameters. For seamless human-machine interaction, it is imperative that this step be performed online and in real-time.

3. LATENT DIRICHLET ALLOCATION

3.1. Background

LDA is a generative probabilistic model for a corpus. A graphical model for the same is shown in Figure 2. The generative process for each document d in a corpus is as follows.

- Choose $\theta \sim Dirichlet(\alpha)$
- For each word w_n in the document -
  - Choose a topic $z_n \sim Multinomial(\theta)$
  - Choose a word $w_n \sim p(w_n|z_n, \beta)$

Here, a document d is a sequence of N words such that $d = (w_1, ..., w_N)$. A corpus is a collection of D such documents, $d_1, ..., d_D$. Each word w_n is defined to be an item from a vocabulary indexed by $\{1, ..., V\}$. θ is a K-dimensional mixture over K latent topics. z_n is a K-dimensional unit-basis vector over topics. The distribution of words over topics is parameterized by a $K \times V$ matrix β. α is a corpus-level parameter sampled once for the entire collection of documents.

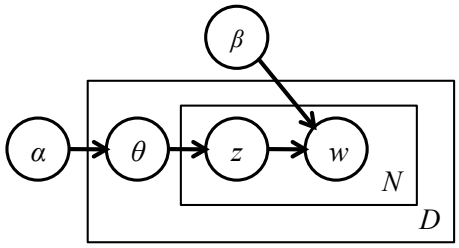

Fig. 2. A graphical model for latent Dirichlet allocation.

Initialize $\phi_{nk} = 1/K$ for all k and n.
Initialize $\gamma_k = \alpha + N/K$ for all k.
repeat
  for $n = 1 : N$ **do**
    for $k = 1 : K$ **do**
      $\phi_{nk} = \beta_{kw_n} exp(\Psi(\gamma_k))$
    end for
    Normalize ϕ_n.
  end for
  $\gamma = \alpha + \sum_{n=1}^{N} \phi_n$
until convergence

Fig. 3. A variational approximation algorithm for LDA.

Parameters α, β and K are estimated and fixed during training. In the recognition step, LDA aims at inferring the latent variables θ and z given the words (observations) w for each document, i.e. $p(\theta, z|w, \alpha, \beta)$. Exact inference in LDA is intractable. A variational approximation method is described in [7]. Briefly, the posterior is modeled by a variational distribution $q(\theta, z|\gamma, \phi)$. γ and ϕ are free variational parameters, iteratively used to minimize the Kullback-Leibler (KL) divergence between q and the posterior, as in (1).

$$(\gamma^\star, \phi^\star) = \arg\min_{(\gamma,\phi)} D(q(\theta, z|\gamma, \phi)||p(\theta, z|w, \alpha, \beta)) \quad (1)$$

An outline of the algorithm to infer (γ, ϕ) and consequently (θ,z) is described in Figure 3. Here, Ψ denotes the digamma function obtained by taking the first-derivative of a log-gamma function.

3.2. Application to Emotion Recognition

For emotion recognition, the speech counterparts of words, topics and documents used in LDA are as follows. We define a speech utterance as the equivalent of a single document. By applying sliding and overlapping windows every 10ms, followed by feature extraction, each utterance is represented as a temporal sequence of p-dimensional feature vectors. These features still lie in $\mathbb{R}^p$ and are continuous-valued. To discretize the feature space, VQ is performed using a codebook of V feature vectors. Thus, each utterance (document) is a collection of discrete, vector quantized features (words).

Training, which is performed offline, involves the estimation of LDA model parameters α and β from the training portion of the corpus via an Expectation-Maximization (EM) algorithm. The posterior distribution over hidden topics, θ, for each utterance is also inferred during this step. Based on the clustering property of LDA, speech utterances with similar emotional content have an approximately similar distribution over θs, while the opposite holds true for emotions belonging to different categories. An SVM using a linear kernel is chosen to train a classifier with θ as the input feature vector.

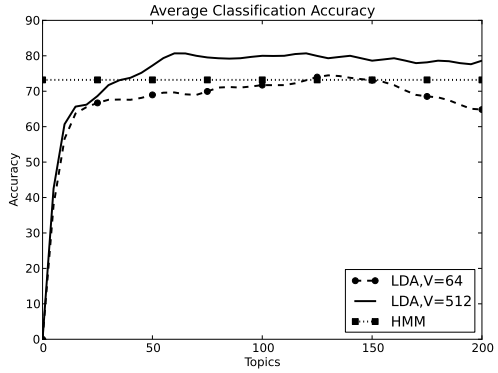

Fig. 4. Average classification accuracy using LDA for vocabulary size $V = 64$ and 512.

The estimated LDA parameters are used to infer θs for unseen data followed by recognition using the trained SVM parameters.

4. IMPLEMENTATION RESULTS

For evaluation of our proposed approach, the freely available German emotional speech database (EMO-DB) [9] is used. It consists of non-spontaneous, acted emotions by 10 speakers, 5 male and 5 female. Each utterance is identified by a single emotion belonging to one of seven categories - *neutral (N)*, *happiness (H)*, *anger (A)*, *fear (F)*, *sadness (S)*, *boredom (B)* or *disgust (D)*. Only 493 utterances with a minimum of 80% human recognition accuracy and 60% naturalness are chosen for our experiments. Feature extraction and post-processing is performed according to the procedure described in section 2, resulting in a 13-dimensional feature vector for every 25ms of speech. Training is performed using 70% of the total utterances and the algorithm is evaluated on the remaining utterances.

4.1. LDA-based Recognition Results

HMMs are used as the baseline to evaluate the proposed approach. An HMM, with 6 states and 2 mixtures per state, is trained for each emotion category using the HTK Speech Recognition Toolkit [10]. The average recognition accuracy is recorded as 73% for this case. In the case of LDA, the number of topics, K, and the vocabulary size, V, are manually set and jointly affect recognition performance. A multi-class, linear kernel-based SVM is trained using a one-versus-rest approach for classification.

Figure 4 shows the variation of recognition accuracy with K for different values of V, averaged over 50 independent runs. For a small-sized vocabulary such as $V = 64$, LDA performs worse than HMM for most values of K, while approaching the baseline accuracy in certain cases. For $V = 512$, LDA always performs better than HMM once K exceeds 50 topics. For K less than 50, the topics fail to efficiently capture the statistical information between features resulting in poorer performance than HMMs. As the vocabulary size V increases, VQ distortion reduces, thus, leading to improved partitioning. This is evident from the improvement in recognition performance as V increases from 64 to 512. For $V = 64$, the maximum accuracy of 74.1% is achieved when $K = 130$; for $V = 512$, the maximum accuracy is higher at 80.7% and is achieved when $K = 60$. The latter case marks a 10.54% improvement over HMM-based recognition. By ignoring the temporal structure, LDA is able to model higher-order dependencies in data compared to HMMs, thus providing better recognition performance.

Table 2. Normalized confusion matrix for LDA

True	Recognized Emotion						
	N	A	H	F	S	B	D
N	**0.64**	0	0	0	0.13	0.23	0
A	0	**1.00**	0	0	0	0	0
H	0.05	0.25	**0.60**	0.05	0	0	0.05
F	0.12	0.12	0	**0.70**	0	0.06	0
S	0.06	0	0	0	**0.94**	0	0
B	0.18	0	0	0	0.05	**0.77**	0
D	0	0	0	0	0	0	**1.00**

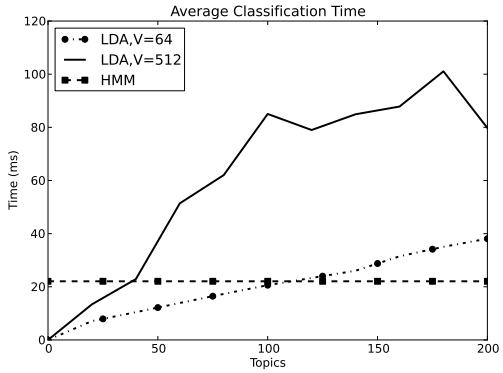

Fig. 5. Average classification time (software) for an utterance of duration 1.5s for vocabulary size $V = 64$ and 512.

Table 2 describes the confusion matrix between different emotions for our approach. If emotions are grouped into two distinct categories based on high (A, H, F) and low (N, S, B, D) arousal, we see from Table 2 that misclassification is more prominent within a group than across groups. For example, 30% of happiness-related utterances are classified as anger and fear, whereas only 10% of the utterances are classified as low-arousal emotions such as neutral or disgust. These results are consistent with findings in previous works [11].

4.2. Software Implementation

The feature extraction, post-processing and LDA routines are implemented in *C* on a Lenovo laptop equipped with an Intel i7 2.7 GHz quad-core processor and 4 GB RAM. OpenMP [12], a freely available software for parallel computing, is used to optimize LDA and achieve speed-up by a factor of 10. A timing comparison of LDA and HMM for the task of classifying 145 utterances is shown in Figure 5. The time complexity of LDA is approximately linear in the number of topics and increases with the vocabulary size as well. For the classification of an utterance of average duration 1.5s, the HMM-based approach takes 22.05ms. In contrast, LDA takes 27.5ms for $V = 64$, $K = 130$ and 55ms for $V = 512$, $K = 60$, while achieving a 10.54% improvement in recognition accuracy. The trade-off between computational complexity and recognition accuracy allows us to adjust the system performance based on the available resources and application requirements.

Table 3. Software and FPGA processing time of proposed system with $V = 512$ and $K = 60$.

	Software (ms)	FPGA (ms)
Feature extraction	322.1 (82%)	1.26 (70%)
Post processing	15.7 (4%)	0.40 (22%)
LDA	55 (14%)	0.15 (8%)
Total(ms)	392.8	1.81

Table 3 lists the processing time for the software implementation. Feature extraction, which involves pre-processing, Hamming windowing and FFT followed by Mel-bank and discrete cosine transform (DCT), accounts for 82% of the total time compared to just 14% for LDA.

4.3. FPGA-based Hardware Implementation

As the size of a database grows, V and K will grow accordingly, thus increasing the processing time. We consider an FPGA-based parallel implementation for handling real-time performance of such systems.

The hardware architecture for the proposed algorithm consists of two main blocks: feature extraction and LDA inference. Both blocks are implemented using Verilog HDL and synthesized on Xilinx Virtex-5 device (XC5VSX240T). The design is verified using Modelsim. For word lengths of 16, 20 and 24 bits, the corresponding classification error rates are 2%, 0.5% and 0% respectively. Based on these results, a 24-bit fixed-point representation is chosen for the FPGA implementation.

Table 4 summarizes the FPGA resource utilization for the feature extraction step. The FIR filter and FFT routines are implemented using Xilinx IP cores. The Mel-bank transform and DCT multiplications are implemented using DSP slices. The utilization is fairly low (8%). A pipelined implementation of feature extraction for 25ms of speech (1 word) takes 841 cycles, while post-processing takes 265 cycles. With a system clock rate of 100 MHz, the total processing period for feature extraction from 1.5s of speech (150 words) is $T_{fe} = 8.41\mu\text{s}\times 150 = 1.261\text{ms}$ and that of post-processing is $T_{pp} = 2.65\mu\text{s}\times 150 = 0.397\text{ms}$.

The LDA inference algorithm described in Figure 3 is implemented by a multiple processing element (PE) architecture, where each PE is assigned to one topic. For the 60-topic system studied here, there are 60 PEs. The critical step of LDA inference is to obtain the digamma factor $\Psi(\gamma)$. One straightforward method is to use a look up table (LUT). In our experiments, $\gamma \in [0.0001, 500]$ and for a resolution of $2^{-14} (< 0.0001)$, the size of LUT is $500 \times 2^{14} = 8$MB. This exceeds the storage space available on the FPGA chip. Alternatively, a Taylor series approximation is used as described in [13]. Since the digamma computations are identical for all topics, we implement only one such calculation engine and house it in a central unit (CU). To avoid access conflicts, we stagger processing times of the PEs. The other computations in each PE such as division and exponential functions are implemented using Xilinx CORDIC IP core and multiplications are implemented using DSP slices.

Resource utilization for the LDA inference engine is shown in Table 5. Each PE occupies 317 (0.8%) slices and CU occupies 4,183 (11%) slices, thus the total occupied slices is 23,203 (62%). For an utterance with 150 words, one LDA iteration takes 295 cycles. Thus, the total processing period for LDA inference (50 iterations) is $T_{LDA} = 2.95\mu\text{s}\times 50 = 0.147\text{ms}$. Finally, the total processing time for an utterance of duration 1.5s is $T_{fe} + T_{pp} + T_{LDA} = 1.805\text{ms}$. Table 3 compares the computation times of the software and FPGA implementations. We see that for the $V = 512$, $K = 60$ system, the FPGA implementation is more than 200 times faster than the software implementation.

Table 4. Resource utilization for feature extraction.

Unit	Occupied slices	Slice Reg.	Slice LUTs	Block RAM	DSP
FIR	153	163	111	0	1
Ham	0	0	1	1	1
FFT	1729	1854	1723	3	10
Mel	897	960	6400	1	40
DCT	268	288	1920	1	12
Total	3047 (8%)	3265 (2%)	10155 (6%)	6 (1%)	64 (6%)

Table 5. Resource utilization for LDA inference engine.

Unit	Occupied slices	Slice Reg.	Slice LUTs	Block RAM	DSP
PE	317	1057	977	1	1
CU	4183	13925	7468	1	9
Total	23203 (62%)	77345 (52%)	66088 (44%)	61 (12%)	69 (7%)

For large databases, V and K will be larger. While the feature extraction time will remain the same, the classification time will increase linearly. For fixed K, as V increases, the time for post-processing and LDA will increase linearly. For fixed V, as K increases, only the LDA processing time increases linearly, others are unchanged. Finally, even for large $V = 1024$ and $K = 180$, the processing time of the FPGA based system is estimated to be only 2.65ms.

5. RELATION TO PRIOR WORK

LDA is primarily intended for modeling and classifying documents [7], though, it has also been used for diverse tasks such as object recognition [14], image annotation [15] and human activity recognition [16]. Experimental results on EMO-DB in [5] and [6] report an average classification accuracy of 73% and 76% for HMM and GMVAR-based methods respectively. In comparison, LDA neglects the temporal information and yields an average accuracy of 80.7%. Previously, FPGA implementations have been explored for speech recognition using HMMs [17, 18]. To the best of our knowledge, our work is the first attempt at implementing LDA and the resulting framework for real-time emotion recognition on an FPGA platform.

6. CONCLUSIONS

In this paper, a complete speech-based emotion recognition framework using LDA and its implementation was presented. LDA was adapted for the purpose of modeling emotional content. The software implementation had timing performance comparable to the HMM-based approach, while achieving a 10.54% improvement in accuracy. An FPGA-based implementation of the proposed system for $V = 512$ and $K = 60$ was able to identify emotions in an utterance of duration 1.5s in 1.8ms. The results presented also show that the proposed system will be capable of real-time emotion recognition even for large databases. Future work will be directed towards benchmarking our approach on diverse and larger databases as well as modifying LDA for joint audio-visual emotion recognition.

7. REFERENCES

[1] R. Cowie and R.R. Cornelius, "Describing the emotional states that are expressed in speech," *Speech Communication*, vol. 40, no. 1, pp. 5–32, 2003.

[2] M. Shah, B. Mears, C. Chakrabarti, and A. Spanias, "Lifelogging: Archival and retrieval of continuously recorded audio using wearable devices," *IEEE International Conference on Emerging Signal Processing Applications*, pp. 99–102, 2012.

[3] M. Shah, B. Mears, C. Chakrabarti, and A. Spanias, "A top-down design methodology using virtual platforms for concept development," *13th International Symposium on Quality Electronic Design*, pp. 444–450, 2012.

[4] M. Grimm, K. Kroschel, E. Mower, and S. Narayanan, "Primitives-based evaluation and estimation of emotions in speech," *Speech Communication*, vol. 49, no. 10, pp. 787–800, 2007.

[5] B. Schuller, B. Vlasenko, F. Eyben, G. Rigoll, and A. Wendemuth, "Acoustic emotion recognition: A benchmark comparison of performances," *IEEE Workshop on Automatic Speech Recognition & Understanding*, pp. 552–557, 2009.

[6] M.M.H. El Ayadi, M.S. Kamel, and F. Karray, "Speech emotion recognition using Gaussian mixture vector autoregressive models," *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 4, pp. 957–960, 2007.

[7] D.M. Blei, A.Y. Ng, and M.I. Jordan, "Latent Dirichlet allocation," *The Journal of Machine Learning research*, vol. 3, pp. 993–1022, 2003.

[8] D. Ververidis and C. Kotropoulos, "Emotional speech recognition: Resources, features, and methods," *Speech Communication*, vol. 48, no. 9, pp. 1162–1181, 2006.

[9] F. Burkhardt, A. Paeschke, M. Rolfes, W. Sendlmeier, and B. Weiss, "A database of German emotional speech," *Proceedings of Interspeech*, pp. 1517–1520, 2005.

[10] S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, "The HTK book," *Cambridge University Engineering Department*, vol. 3, 2002.

[11] T.L. Nwe, S.W. Foo, and L.C. De Silva, "Speech emotion recognition using hidden Markov models," *Speech Communication*, vol. 41, no. 4, pp. 603–623, 2003.

[12] L. Dagum and R. Menon, "OpenMP: an industry standard API for shared-memory programming," *IEEE Computational Science & Engineering*, vol. 5, no. 1, pp. 46–55, 1998.

[13] M.J. Beal, "Variational algorithms for approximate Bayesian inference (PhD thesis)," *The Gatsby Computational Neuroscience Unit, University College London*, pp. 65–66, 2003.

[14] B.C. Russell, W.T. Freeman, A.A. Efros, J. Sivic, and A. Zisserman, "Using multiple segmentations to discover objects and their extent in image collections," *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 1605–1614, 2006.

[15] C. Wang, D. Blei, and F.F. Li, "Simultaneous image classification and annotation," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1903–1910, 2009.

[16] T. Huynh, M. Fritz, and B. Schiele, "Discovery of activity patterns using topic models," *Proceedings of the 10th International Conference on Ubiquitous Computing*, pp. 10–19, 2008.

[17] E.C. Lin, K. Yu, R.A. Rutenbar, and T. Chen, "Moving speech recognition from software to silicon: the in silico vox project," *Proceedings of Interspeech*, pp. 2346–2349, 2006.

[18] E.C. Lin, K. Yu, R.A. Rutenbar, and T. Chen, "A 1000-word vocabulary, speaker-independent, continuous live-mode speech recognizer implemented in a single FPGA," *Proceedings of the 15th International Symposium on Field programmable gate arrays*, vol. 18, no. 20, pp. 60–68, 2007.