N-Best Rescoring by Phoneme Classifiers using Subclass AdaBoost Algorithm

Hiroshi Fujimura, Yusuke Shinohara, Takashi Masuko

Corporate Research and Development Center, Toshiba Corporation

{hiroshi4.fujimura, yusuke.shinohara, takashi.masuko}@toshiba.co.jp

Abstract

This paper proposes a novel technique to exploit discriminative models with subclasses for speech recognition. Speech recognition using discriminative models has attracted much attention in the past decade. However, most discriminative models are still based on tree clustering results of HMM states. On the contrary, our proposed method, referred to as subclass AdaBoost, jointly selects optimal data split and weak discriminators in each iteration of the training process, and forms a weak classifier as a composite of these weak discriminators. As a result, a strong discriminator robust to a variety of subclasses is constructed without explicit clustering in advance. In the experiment, the subclass AdaBoost is applied to phoneme classification, and N-best hypotheses are rescored using the subclass AdaBoost phoneme classifiers. Experimental results show that the proposed method reduces word errors by over 10% relatively in a continuous speech recognition task.

Index Terms: speech recognition, AdaBoost, discriminative model

1. Introduction

Many applications that exploit automatic speech recognition (ASR) demand better performance for large vocabulary ASR. The ASR system has to discriminate the target phonemes from others because only a few different phonemes are the clues to discriminate similar words. Therefore, we focus on phoneme discrimination in speech recognition.

The hidden Markov model (HMM) with Gaussian Mixture Model (GMM) is the principal technique for automatic speech recognition. Many efforts have been made to improve performance of phoneme discrimination for GMM-HMM. Some studies succeeded in the improvement of the accuracy by Minimum Phone Error (MPE) and Maximum Mutual Information (MMI) discriminative training for GMM-HMM [1] [2] [3]. These are based on GMM-HMM. On the other hand, some studies try to replace GMM by a discriminative model directly. Ganapathiraju *et al.* [4] combined HMM and the support vector machine (SVM) which is one of the effective discriminative models. Their technique applies SVMs to phoneme segments obtained from the alignment of HMM, and N-best hypotheses are rescored by scores of SVMs. Padrell-Sendra *et al.* [5] applied SVM to the frame-level discrimination, and obtained better performance than a standard GMM-HMM. SVMs are constructed for three classes per monophone. Ragni and Gales [6] exploited generative model and discriminative model with structured discriminative models. Recently HMM with Deep Neural Network (DNN-HMM) showed remarkable improvement to GMM-HMM [7]. We also proposed AdaBoost rescoring for the second pass of GMM-HMM [8], which improved the isolated word recognition performance for the discrimination of monophones.

For speech recognition, the feature of a monophone changes depending on the contexts. Hence, a model is constructed for the triphone unit. However many speech recognition systems use tied-state triphones which are tied by the context clustering as subclasses because the number of triphones is large, and the samples for model training become sparse. Therefore, the context clustering for triphones is the key essence for phone modeling. For SVM phone modeling [4] [5], SVMs are constructed for monophones. DNN-HMMs [7] are modeled for tied-state triphones which are constructed by tree clustering with Maximum Likelihood criterion. However the tied-state triphones from tree clustering are not optimized for the discriminator. In some studies, discriminative criterion is used for creating the tied-state triphones. Wiesler *et al.* [9] achieved 10% relative improvement by tree clustering with the minimization of the classification error. Sim [10] proposed probabilistic state clustering with Conditional Random Field for HMM with GMM. These studies create tied-state triphones with the discriminative criteria. They can be applied to DNN-HMM and other discriminative methods. However they are not optimized for phoneme discriminators themselves. Therefore, it is desired that the tied-state triphones are directly optimized for phoneme discriminators.

In our method, this is achieved by AdaBoost [11] [12]. In image recognition field, some ideas of subclass AdaBoost [13] [14] [15] are proposed. In these studies, subclass AdaBoost is applied after the estimation of subclasses for input vectors. Hence, they are not the techniques which exploit subclass label such as triphone contexts which are given without estimation. In many speech recognition systems, subclass attribution, which is the key of split, is given such as phoneme contexts in advance. In our method, weak classifiers of AdaBoost are designed for exploiting subclass labels such as phoneme contexts. The training samples of AdaBoost is split into subclasses by the phoneme contexts, and the best discriminators are selected for the subclasses for each weak classifier training step of AdaBoost. It is iterated for all prepared subclass-splits. In this step, the subclasses and the discriminators which have the least error for training samples are selected as the weak classifier. The AdaBoost which is constructed by such weak classifiers is jointly optimized for subclasses and the discriminative criterion, implicitly.

As a first try of subclass AdaBoost application, classification scores are calculated using subclass AdaBoost classifiers for the phoneme segments of N-best hypotheses given by HMMs, and the N-best hypotheses are re-ordered based on the classification scores. The subclass AdaBoost classifies whether the phoneme of a segment is correct or not as a binary classifier.

The reminder of this paper is organized as follows: Section 2 introduces the proposed subclass AdaBoost algorithm. Section 3 introduces the proposed rescoring method by AdaBoost. Section 4 shows experimental setups and results of the proposed rescoring method. Conclusions are presented in section 5.

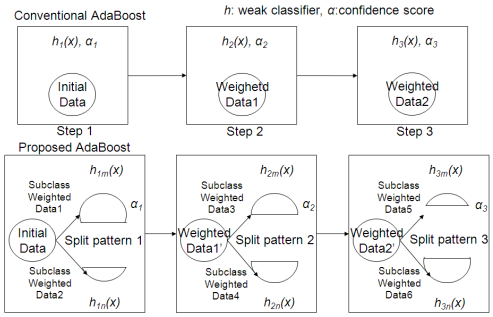

Figure 1: *The overview of the proposed AdaBoost training*

2. Subclass AdaBoost

2.1. Algorithm

The overview of the subclass AdaBoost is shown in Fig.1. The difference between conventional and proposed AdaBoost is mainly data split at each weak classifier training. Each weak classifier is optimized by data split and the classification error. For the purpose of the data split, some rules are prepared in advance before the subclass AdaBoost training. In the speech recognition field, the rule is defined by the triphone context as the conventional tree clustering uses it.

Assume that there are N training samples $(x_1, y_1), ..., (x_N, y_N)$, where x_i and y_i denote the feature vector and the class label of sample i. Normal AdaBoost classifiers are trained by the following steps:

- Step 1. Initialize sample weight distribution $D_0(i)$ by the following equation:

$$D_0(i) = \begin{cases} \dfrac{1}{2\sum_{j:y_j=1} 1}, & y_i = 1, \\ \dfrac{1}{2\sum_{j:y_j=-1} 1}, & y_i = -1. \end{cases} \tag{1}$$

- Step 2. Train a weak classifier $h_t(x_i)$ minimizing error rate ε_t on sample weight distribution D_t,

$$\varepsilon_t = \sum_{i:y_i \neq h_t(x_i)} D_t(i). \tag{2}$$

- Step 3. Compute voting weight α_t as

$$\alpha_t = \frac{1}{2} \log \frac{1-\varepsilon_t}{\varepsilon_t}. \tag{3}$$

- Step 4. Recompute sample weight distribution as

$$\hat{D}_{t+1}(i) = D_t(i) \exp(-\alpha_t y_i h_t(x_i)). \tag{4}$$

- Step 5. Normalize the summation of sample weights to 1

$$D_{t+1}(i) = \frac{\hat{D}_{t+1}(i)}{Z_{t+1}}, \tag{5}$$

where

$$Z_{t+1} = \sum_{i=1}^{N} \hat{D}_{t+1}(i). \tag{6}$$

- Step 6. Iterate from step 2 to step 5 T times, and obtain T weak classifiers.

  Finally the strong classifier $H(x)$ is obtained by weighted sum of T weak classifiers as

$$H(x) = \text{sign}\left\{ \sum_{t=1}^{T} \alpha_t h_t(x) \right\}, \tag{7}$$

Algorithm 1 The training step of the t-th iteration

There are 3 binary questions for an example:
Binary question 1 :previous phoneme is "a"; $true$ or $false$,
Binary question 2 :previous phoneme is "i"; $true$ or $false$,
Binary question 3 :previous phoneme is "u"; $true$ or $false$.

for $k = 1$ to 3 **do**
- Apply "Binary question k" to samples.
- Samples divided into subclass "k:true" and "k:false".
- Find the best discriminators $h_k^{(true)}$ and $h_k^{(false)}$ to minimize $\varepsilon_t^{(k)}$,

$$\varepsilon_t^{(k)} = \sum_{\substack{i:z_i=true \\ y_i \neq h_k^{(true)}(x_i),}} D_t(i) + \sum_{\substack{i:z_i=false \\ y_i \neq h_k^{(false)}(x_i),}} D_t(i).$$

end for
- Find $k_{min} (\in \{1, 2, 3\})$ to minimize $\varepsilon_t^{(k)}$.
- Calculate α_t using $\varepsilon_t^{(k_{min})}$
- Calculate the distribution D_{t+1} using $\varepsilon_t^{(k_{min})}$.
- Restore k_{min}, α_t, $h_{k_{min}}^{(true)}$, $h_{k_{min}}^{(false)}$ as the weak classifier parameters of this step.
- Go to the $(t+1)$-th iteration.

  where each weak classifier outputs 1 (true) or -1 (false) by comparing a threshold and values of a selected dimension of the segment feature vectors.

In our proposed method, Step 2 is different from normal AdaBoost training step. First some classification rules are prepared in advance. Let K denote the classification rules. Samples in D_t are divided into subclasses by a rule $k \in K$. Let M_k denote the subclasses constructed by the split rule k. The best weak classifier $h_k^{(m)}$ (m: class label for subclasses M_k) is detected for each subclass m. The total error ε_k is calculated by sum of subclass errors. The minimum total error $\varepsilon_{k_{min}}$ is found in all $k \in K$. Finally, the subclasses $M_t = M_{k_{min}}$ and the best classifiers $h_t^{(m)}$ are restored as they minimize the total error. The new Step2 is the following:

- Step 2'-1. For all $k (\in K)$, the best weak classifiers $h_k^{(m)}$ for subclasses M_k split by the rule k are detected.

$$\varepsilon_t^{(k)} = \sum_{m \in M_k} \sum_{\substack{i:z_i=m \\ y_i \neq h_k^{(m)}(x_i),}} D_t(i), \tag{8}$$

  where z_i is the label of ith sample.
- Step 2'-2. Find k to minimize the $\varepsilon_t^{(k)}$.

$$k_{min} = \arg\min_k \varepsilon_t^{(k)} \tag{9}$$

- Step 2'-3. Decide $\varepsilon_t = \varepsilon_t^{(k_{min})}$, $M_t = M_{k_{min}}$ and $h_t^{(m)} = h_{k_{min}}^{(m)}$, where $m \in M_t$.

For the application of the speech recognition, the split rule is the triphone context which is "a-*" or not for example. The simplest way is to split by binary questions which are usually applied to tree clustering by ML estimation. The binary questions are prepared in advance, and all questions are applied to data split at each weak classifier training. The best weak classifiers in an iteration are decided by total errors after all patterns by the questions are tried. A sample of the algorithm is shown in Algorithm1. To do classification using a subclass AdaBoost classifier, given an input vector and left and right contexts, for each weak classifier, the input vector is assigned to either of the two subclasses by the binary question, and the discriminator correspoiding to the subclass is applied to the input vector.

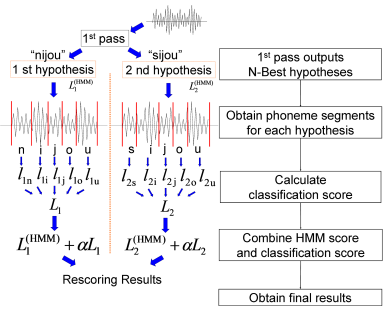

Figure 2: *The flowchart of the second pass*

3. Rescoring Using AdaBoost Classifiers

3.1. Overview of the Rescoring Process

The subclass AdaBoost can be applied to many applications. An example is shown for applying the subclass AdaBoost to a speech recognition system [8]. This section describes the proposed method to rescore the N-best hypotheses using subclass AdaBoost phoneme classifiers as [16]. Our method consists of the first pass and the second pass. The first pass outputs N-best hypotheses with scores by a conventional speech recognition system using HMM. The process of the second pass is shown in Fig.2. First, phoneme segments are obtained by forced alignment of phoneme HMMs for each hypothesis. Then, segment features are extracted from each phoneme segment, and classification scores are calculated by subclass AdaBoost classifiers using the segment features. For the subclass AdaBoost, an input label has left and right contexts. In Fig.2, the input labels of the hypothesis word "nijou" are "sil-n+i, n-i+j, i-j+o, j-o+u, o-u+sil", where "sil" means silence phoneme. Finally, classification scores are added to the N-best scores and N-best hypotheses are re-ordered.

3.2. Segment Feature

Although lengths of phoneme segments are variable, fixed-length features are desirable for most classifiers. Hence, fixed-length features are extracted from phoneme segments with variable length. Alignment information of HMM states is exploited for this purpose. Figure 3 shows the segment feature used for AdaBoost. Assume without loss of generality that phoneme HMMs are three-state left-to-right models and frame-based features are extracted in advance. In this report, LPC-spectrum and MFCCs are extracted as frame-based features in advance. First, the phoneme segment for time and LPC-spectrum plane is normalized to average 0 and variance 1. Then, mean and variance are calculated in each Mel scale block of the phone segment. The segment is extended by one frame for the start and end of the state alignment because frames of calculating variance are allocated. Second, mean and variance are calculated in each Mel scale block of each state segment. Third, mean is calculated at each dimension in the phone segment for MFCCs. Forth, mean is calculated at each dimension in each state segment for MFCCs. Finally, all vectors from LPC-spectrum and MFCCs are concatenated to form a fixed-length vector expressing the phoneme segment information.

3.3. Classification Score

Phoneme classifier is constructed for each phoneme by an AdaBoost algorithm. The process of the calculation of classification score is explained using the conventional AdaBoost case. Each phoneme classifier discriminates whether the given segment is the phoneme or not, and outputs the AdaBoost score for

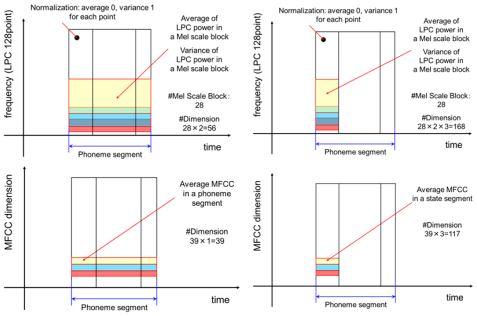

Figure 3: *Fixed length feature extraction from variable length segment*

the segment. As described in [17], the AdaBoost score $S_p(x)$ of a phoneme p for segment feature x is derived from the following equation,

$$S_p(x) = \frac{1}{\sum_{t=1}^{T} \alpha_t^{(p)}} \sum_{t=1}^{T} \alpha_t^{(p)} h_t^{(p)}(x). \tag{10}$$

Note that the range of $S_p(x)$ is $-1 \leq S_p(x) \leq 1$ due to the normalization. Classification scores are calculated from the AdaBoost scores. First the classification score is calculated at each phoneme segment in all N-best hypotheses. Classification score l_i of i-th phoneme segment in a hypothesis is obtained by the following equation:

$$l_i = \log \frac{S_{ip_i} + 1}{\sum_{k=1}^{|P|} (S_{ik} + 1)}, \tag{11}$$

where p_i denote the phoneme of the i-th segment in the hypothesis, and S_{ik} and $S_i p_i$ are AdaBoost scores for phoneme k of i-th segment and for phoneme p_i of i-th segment, respectively. In order to guarantee the normalized score is positive, a constant 1 is added to S_{ik} and $S_i p_i$. Note that the score l_i can be considered as a logarithm of a posteriori probability based on AdaBoost scores.

The classification score L for a hypothesis is obtained by averaging l_i in the hypothesis,

$$L = \frac{1}{K} \sum_{i=1}^{K} l_i, \tag{12}$$

where K is the number of phonemes in this hypothesis. Finally, the rescoring score L_{re} is obtained by weighted sum of L and HMM score $L^{(\mathrm{HMM})}$ of the hypothesis,

$$L_{\mathrm{re}} = L^{(\mathrm{HMM})} + \alpha L, \tag{13}$$

and the N-best hypotheses are re-ordered using the score L_{re}.

4. Experiments

4.1. Experimental Setup

4.1.1. Evaluation Data

Experiments were conducted using Japanese speech database. Two databases are evaluated. One is the travel domain corpus which includes utterances by 21 speakers. Each speaker utters 200 sentences about the travel domain. The data set is divided into development set of 5 speakers and evaluation set of 16 speakers. Another is the Corpus of Spontaneous Japanese (CSJ) [18] which is a Japanese standard evaluation set for the continuous speech recognition.

4.1.2. HMM training

The HMMs were trained by 500-hour CSJ speech data which doesn't include test set data. The basic structure of the HMM is

Table 1: *Training and Evaluation Data*

Language	Japanese
Acoustic Model Training Data	CSJ corpus 500 hours
Language Model Training Data	CSJ corpus and Travel domain text are mixed
#Evaluation utterances	Travel domain 3200 (16speakers) CSJ testset3 2484 (10speakers)
Evaluation Task	30 thousands continuous words recognition
Basic Feature	12-dimensional MFCCs + normalized power, their Δs and $\Delta\Delta$s

three-state continuous density triphones that share 3000 states with 32 Gaussian mixture components. All triphones have a simple left-to-right topology. A feature vector for the HMM consisted of 12 MFCC (Mel-frequency cepstrum coefficient), a normalized power, and their Δs and $\Delta\Delta$s (totally 39 dimensions). HTK [19] was used for the HMM training. HLDA (heteroscedastic linear discriminant analysis) [20] without nuisance dimensions and MPE (minimum phone error) training [1] were applied to the HMM after the MLE training.

4.1.3. AdaBoost Training

Basically, the training process of the subclass AdaBoost is the same as the conventional AdaBoost. AdaBoost was trained using the same data as HMMs'. The feature for AdaBoost were obtained by the method described in Sec.3.2 based on forced alignment of HMMs. LPC-spectrum is calculated from 17th order LPC. MFCCs are the same as the HMM training. Consequently, there were 380 dimensions for AdaBoost features. AdaBoost classifier were constructed for each phoneme, and each AdaBoost classifier discriminates if a given segment is the phoneme or not. The weak classifier was constructed by a threshold decision for a dimension [12]. The tied-state triphone models were used for forced alignment to obtain phoneme segments of the training data with the correct phoneme labels. Error phoneme segments were extracted from the lattices which were created in HMM-MPE process. The positive labels are attached to the target phoneme labels of the correct phoneme labels, and the negative labels are attached to the other phoneme labels of the correct phoneme labels and error phoneme segments from the lattices. The number of phonemes excluding silence was 38 in the experiments. For Subclass AdaBoost, the data at each weak classifier was split by binary questions for triphone contexts. The number of weak classifier steps was 1200 for the conventional AdaBoost and 600 for the subclass AdaBoost, respectively.

4.1.4. Language Model Training

The standard 3-gram language model was constructed using CSJ text corpus and travel domain text corpus. The text corpus for evaluation test was not included in the training. The portion of travel domain text is the same as CSJ's.

4.2. Phoneme Classification Experiment

A phoneme classification experiment was conducted to confirm the classification performance for the subclass AdaBoost. First, the evaluation utterances were aligned by the tied-state triphone HMMs with the correct labels. Then for each segment, scores in Eq.(10) were calculated using AdaBoost models, and the phoneme label with the highest score was output as a classification result. For the subclass AdaBoost, correct left and right contexts were give as an input label. Table 2 shows the results of conventional AdaBoost and subclass AdaBoost. It was evaluated using CSJ testset3. It shows that the performance of the subclass AdaBoost is highly better than the conventional AdaBoost. The error reduction rate is 53.49%.

Table 2: *The classification performance by classification error rate [%]*

	Conventional AdaBoost	Subclass AdaBoost
CSJ tesetset3	17.46	8.12

Table 3: *The performance of rescoring methods by Word Error Rate [%]*

	HMM	Rescore(conventional)	Rescore(subclass)
Travel	14.00	13.02	12.42
CSJ testset3	19.08	18.76	18.70

4.3. Rescoring Experiment

Performance of the proposed rescoring technique was evaluated on the continuous word recognition tasks. In this experiment, the first pass output 1000-best hypotheses, and they were rescored by the AdaBoost scores in the classification scores in Eq.(12) in the second pass. Table 3 shows the performance of the rescoring. "HMM" denotes results of the first pass, "Rescore(conventional)" denotes results of rescoring by the classification scores calculated from conventional AdaBoost scores. "Rescore(subclass)" denotes results from subclass AdaBoost scores. Language model weight and Coefficient α in Eq.(13) are adjusted to maximize the performance on development set for the travel corpus. For the CSJ corpus, they are adjusted to maximize the performance on the evaluation set itself. Accordingly, the rescoring results are better than the 1st pass results for both of tasks. For the travel corpus, the word error reduction rate was 11.29% by subclass AdaBoost, which is better than the performance of the conventional AdaBoost. However the performance of the rescoring for CSJ is not improved much. Furthermore, the difference of performance between conventional and subclass AdaBoosts is very small although the difference of classification performance is large. CSJ is the spontaneous speech database, which includes a lot of ambiguous speech utterances. The rescoring method by phoneme segments may not match the solution of the spontaneous speech recognition. This is the future work for us.

5. Conclusions

We proposed a novel technique to exploit discriminative models with subclasses for speech recognition. Our method is subclass AdaBoost which does not need splits by any clustering method in advance. Subclass AdaBoost can automatically optimize the discriminator with triphone contexts. This method can be applied to many applications. As a first try, a phoneme classifier is constructed for each phoneme by the subclass AdaBoost algorithm. Each phoneme classifier discriminates whether the given segment is the phoneme or other phonemes, and outputs the AdaBoost score for the segment as the classification score. In the classification task, the performance of the subclass AdaBoost is highly better than the conventional AdaBoost. The error reduction rate is 53.49%. Furthermore, experimental results showed that the proposed technique consistently improved the continuous word recognition performance. The word error reduction rate was max 11.29% , when the number of weak classifiers was 600 using the proposed subclass AdaBoost.

6. References

[1] D. Povey and P. C. Woodland, "Minimum phone error and I-smoothing for improved discriminative training," *Proc. ICASSP*, pp.I-105-I-108, April 2002.

[2] D. Povey and P. D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon and K. Visweswariah, "Boosted MMI for Model and Feature-Space Discriminative Training," *Proc. ICASSP*, pp.4057-4060, April 2008.

[3] G. Heigold, R. Schluter and H. Ney, "Modified MPE/MMI in a Transducer-based Framework," *Proc. ICASSP*, pp.3749-3752, April 2009.

[4] A. Ganapathiraju, J. Hamaker, and J. Picone, "Applications of support vector machines to speech recognition," *Signal Processing, IEEE Trans*. Vol. 52, Issue 8, 2004.

[5] J. Padrell-Sendra, D. Martin-Iglesias, and F. Diaz-de-Maria, "Support vector machines for continuous speech recognition," In *Proceedings of the 14th European Signal Processing Conference*, Florence, Italy, 2006.

[6] A. Ragni and M.J.F Fales, "Inference Algorithms for Generative Score-Spaces," *ICASSP*, 2012.

[7] G. Hinton, L. Deng, D. Yu, G. Hahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhouchke, P. Nguyen, T. Sainath and B. Kingsbury, "Deep Neural Networks for Acoustic Modeling in Speech Recognition," *IEEE Signal Processing Magazine*, 29, Nov.2012.

[8] H. Fujimura, M. Nakamura, Y. Shinohara and T.Masuko, "N-Best Rescoring by AdaBoost Phoneme Classfiers for Isolated Word Recognition," *ASRU*, 2011.

[9] S. Wiesler, G. Heigold, M. Numbaum-Thom, R. Schluter and H. Ney, "A Discriminative Splitting Criterion for Phonetic Decision Trees," *INTERSPEECH*, 2010.

[10] Khe Chai Sim, "Probabilistic State Clustering Using Conditional Random Field For Context-Dependent Acoustic Modeling, " *INTERSPEECH*, 2010.

[11] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and Science*, Vol. 55, pp. 119-139, 1997.

[12] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *Proc. CVPR*, pp. 511-518, 2001.

[13] S. Z. Li, L. Zhu, Z.Q. Zhang, A. Blake, H. Zhang, H. Shum, "Statistical Learning of Multi-View Face Detection," *In Proceedings of the 7th European Conference on Computer Vision*, 2002.

[14] C. Huang, H. Ai, Y. Li, S. Lao, "Vector Boosting for Rotation Invariant Multi-View Face Detection," *Proc. ICCV2005*, Vol. 1, 17-21, pp. 446-453 Oct. 2005.

[15] L. Ding and M. Martinez, "Features versus Context: An Approach for Precise and Detailed Detection and Delineation of Faces and Facial Features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 32, No.11, Nov. 2010.

[16] S. M. Siniscalchi, J. Li, C. Lee, "A study on lattice rescoring with knowledge scores for automatic speech recognition," *INTERSPEECH*, 2006.

[17] K. Shutte and J. Glass, "Speech Recognition with Localized Time-Frequency Pattern Detectors," *ASRU*, pp. 341-346, 2007.

[18] T. Kawahara, H. Nanjo, T. Shinozaki and S. Furui, "Benchmark Test for Speech Recognition Using the Corpus of Spontaneous Japanese," *In Proc. ISCA & IEEE WORKSHOP ON SPONTANEOUS SPEECH PROCESSING AND RECOGNITION*, 2003.

[19] S. Young *et al.*, "The HTK Book," Cambridge University Engineering Department, 2009.

[20] N. Kumar and A.G. Andreou, "Heteroscedastic discriminant analysis and reduced rank HMMs for improved speech recognition, " *Speech Communication*, Vol. 26, pp. 283-297, 1998.