IMAGE RECOGNITION BASED ON SEPARABLE LATTICE HMMS USING A DEEP NEURAL NETWORK FOR OUTPUT PROBABILITY DISTRIBUTIONS

Eiji Ichikawa, Kei Sawada, Kei Hashimoto, Yoshihiko Nankaku, and Keiichi Tokuda

Department of Computer Science and Engineering, Nagoya Institute of Technology, Nagoya, Japan

ABSTRACT

This paper proposes an image recognition method based on separable lattice hidden Markov models (SLHMMs) using a deep neural network (DNN) for output probability distributions. The geometric variations of the object to be recognized, e.g., size and location, are essential in image recognition. SLHMMs, which have been proposed to reduce the effect of geometric variations, can perform elastic matching both horizontally and vertically. Gaussian distributions are typical for modeling the output distribution of SLHMMs. However, these distributions may not be sufficient to represent patterns of image regions. Our method integrates SLHMMs and a DNN and can be used to model an image effectively by explicit modeling of the generative process based on SLHMMs and advanced feature classification based on a DNN. image recognition experiments showed that the proposed method improves recognition performance.

***Index Terms*—** image recognition, separable lattice HMMs, deep neural networks, DNN-HMM, DNN-SLHMM

1. INTRODUCTION

In the field of machine learning and pattern recognition, statistical methods have grown in popularity in the last decade. In image recognition, for instance, eigenface methods [1] and subspace methods [2] achieve good recognition performance. However, such statistical methods encounter a problem in terms of geometric variations, i.e., position, size, and rotation, of target objects. One of the major solutions to this problem is the pre-normalization process for geometric variations prior to applying statistical methods. In general, normalization is performed manually or using an empirically developed normalization technique independently of training and recognition. However, normalization is not optimized to solve the classification problem because the normalization criterion is determined heuristically. Therefore, it seems ideal to integrate the normalization process into classifiers and optimize them simultaneously based on the unified criterion. Currently, convolutional neural network (CNN)-based methods, which integrate geometric invariants into model structures, have achieved great success in image recognition [3, 4]. In addition to the structure of the feed-forward neural networks as classifiers, CNNs have geometric invariants based on multiple convolutional and pooling layers. However, since the pooling process is independently performed in each local window, it is difficult to represent global geometric transforms over an entire image.

Another way to integrate the normalization processes into model structures is using hidden Markov models (HMMs) [5, 6]. The geometric normalization is represented by discrete hidden variables, and the normalization process is performed through the calculation of probabilities. Although the extension of HMMs to multiple dimensions generally leads to an exponential increase in the computational complexity, some efficient approximations of likelihood calculation and model structures have been proposed [7–14]. Among them, separable lattice HMMs (SLHMMs) reduce computational complexity while retaining outstanding properties that model two-dimensional data. SLHMMs are feasible models that can perform an elastic matching in both vertical and horizontal directions, making it possible to model invariances to the size and location of an object. One of the advantages of SLHMMs over CNNs is explicit modeling of the generative process, which can represent geometric variations over an entire image. A single Gaussian distribution is usually used as an output distribution corresponding to each state for SLHMMs. However, the ability of a Gaussian distribution is not sufficient to represent patterns of image regions. Therefore, it seems possible to improve the generalization performance of an SLHMM by using a distribution with high expression ability instead of a Gaussian distribution.

Recently, in speech recognition and natural language processing, an integrated model of HMMs and a deep neural network (DNN) has been proposed to model one-dimensional time series data and is called DNN-HMM [15–19]. DNN-HMM estimates the output probability for each state of HMMs using a DNN. Compared to GMM-HMM, which is an integrated model of HMMs and Gaussian mixture models (GMMs) [20, 21], DNN-HMM can estimate the output probability with high accuracy. As a result, DNN-HMM-based methods can properly model one-dimensional time series data and has achieved large improvements over conventional GMM-HMM-based methods for tasks such as speech recognition.

In this paper, we propose an image recognition method based on SLHMMs using a DNN for the output distribution as a model of DNN-HMM extended to two dimensions. The proposed method, called DNN-SLHMM, integrates SLHMMs and a DNN. Therefore, it can represent geometric variations by SLHMMs and estimate the appropriate output probability with the advanced discrimination capability of DNNs. As a result, the recognition performance of DNN-SLHMM-based method is expected to improve compared with conventional methods.

2. SEPARABLE LATTICE HIDDEN MARKOV MODELS

SLHMMs [13, 14] are defined for modeling multi-dimensional data. In the case in which observations are two-dimensional data, e.g., pixel values of an image, observations are assumed to be given on a two-dimensional lattice as

$$\boldsymbol{O} \quad = \quad \{\boldsymbol{O_t} | \boldsymbol{t} = (t^{(1)}, t^{(2)}) \in \boldsymbol{T}\}, \tag{1}$$

where $\boldsymbol{t}$ denotes the coordinates of the lattice in two-dimensional space $\boldsymbol{T}$ and $t^{(m)} = 1, ..., T^{(m)}$ are the coordinates of the m-th dimension for $m \in \{1, 2\}$. In two-dimensional HMMs, observation $\boldsymbol{O_t}$ is emitted from the state indicated by hidden variable $\boldsymbol{z_t} \in \boldsymbol{K}$. The hidden variables $\boldsymbol{z_t} \in \boldsymbol{K}$ can take one of $K^{(1)}K^{(2)}$ states,

which are assumed to be arranged on a two-dimensional state lattice $\boldsymbol{K} = \{(1,1),(1,2),...,(K^{(1)},K^{(2)})\}$.

In SLHMMs, the hidden variables are constrained to be composed of two Markov chains to reduce the number of possible state sequences as

$$\boldsymbol{z} = \left\{\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}\right\}, \tag{2}$$

$$\boldsymbol{z}^{(m)} = \left\{z^{(m)}_{t^{(m)}} | 1 \leq t^{(m)} \leq T^{(m)}\right\}, \tag{3}$$

where $\boldsymbol{z}^{(m)}$ is the Markov chain along with the m-th coordinate and $z^{(m)}_{t^{(m)}} \in \{1,2,...,K^{(m)}\}$. The composite structure of hidden variables in SLHMMs is defined as the product of hidden state sequences: $\boldsymbol{z_t} = (z^{(1)}_{t^{(1)}}, z^{(2)}_{t^{(2)}}) \in \boldsymbol{K}$. This means that the segmented regions of observations are constrained to be rectangles, which allows an observation lattice to be elastic in both vertical and horizontal directions. Figure 1 shows a graphical model of SLHMMs. The joint probability of observation vectors $\boldsymbol{O}$ and hidden variables $\boldsymbol{z}$ can be written as

$$\begin{aligned} P(\boldsymbol{O},\boldsymbol{z}|\boldsymbol{\Lambda}) &= P(\boldsymbol{O},\boldsymbol{z}^{(1)},\boldsymbol{z}^{(2)}|\boldsymbol{\Lambda}) \\ &= \prod_{\boldsymbol{t}} P(\boldsymbol{O_t}|\boldsymbol{z_t},\boldsymbol{\Lambda}) \\ &\times \prod_{m=1}^{2}\left\{P(z^{(m)}_1|\boldsymbol{\Lambda}) \prod_{t^{(m)}=2}^{T^{(m)}} P(z^{(m)}_{t^{(m)}}|z^{(m)}_{t^{(m)}-1},\boldsymbol{\Lambda})\right\}, \end{aligned} \tag{4}$$

where $\boldsymbol{\Lambda}$ is the model parameter, $P(z^{(m)}_1|\boldsymbol{\Lambda})$ is the initial state probability, $P(z^{(m)}_{t^{(m)}}|z^{(m)}_{t^{(m)}-1},\boldsymbol{\Lambda})$ is the state transition probability, and $P(\boldsymbol{O_t}|\boldsymbol{z_t},\boldsymbol{\Lambda})$ is the state output probability.

In image recognition based on SLHMMs, an SLHMM is trained for each class by using the expectation-maximization (EM) algorithm [22, 23], and classification is performed by selecting the class yielding the maximum posterior probability.

$$\begin{aligned} \hat{C} &= \arg\max_{C\in\boldsymbol{C}} P(C|\tilde{\boldsymbol{O}},\boldsymbol{\Lambda}) \\ &= \arg\max_{C\in\boldsymbol{C}} P(\tilde{\boldsymbol{O}}|C,\boldsymbol{\Lambda})P(C), \end{aligned} \tag{5}$$

where $\tilde{\boldsymbol{O}}$ is testing data, $\boldsymbol{C} = \{C_1, C_2, ..., C_N\}$ is class, N is the number of classes, $\hat{C}$ is the classification result, and $P(C)$ is the occurrence probability of class C. If it is assumed that $P(C)$ is constant irrespective of the input image, Eq. (5) becomes

$$\begin{aligned} \hat{C} &= \arg\max_{C\in\boldsymbol{C}} P(\tilde{\boldsymbol{O}}|C,\boldsymbol{\Lambda}) \\ &= \arg\max_{C\in\boldsymbol{C}} \sum_{\boldsymbol{z}} P(\tilde{\boldsymbol{O}},\boldsymbol{z}|C,\boldsymbol{\Lambda}). \end{aligned} \tag{6}$$

From the above equation, the class that obtains the highest likelihood is then chosen as the classification result.

3. SLHMMS USING A DNN FOR OUTPUT PROBABILITY DISTRIBUTIONS

A single Gaussian distribution or GMMs are typical for modeling the output distribution $P(\boldsymbol{O_t}|\boldsymbol{z_t},\boldsymbol{\Lambda})$ for each state in SLHMMs. However, these distributions may not properly express the relationship between image feature and SLHMM states.

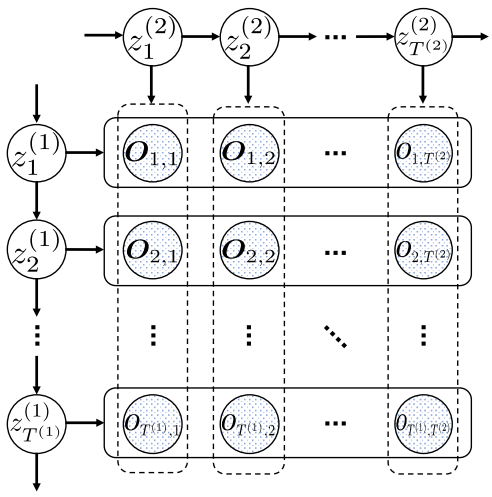

Fig. 1. Graphical model of SLHMMs. Rounded boxes represent group of variables, and arrow to each box represents dependency in regard to all variables in box instead of drawing arrows to all variables.

Currently, GMM-HMM has been used as an integrated model of one-dimensional HMMs and GMMs [20, 21]. GMM-HMM is widely used as an acoustic model in speech recognition and can achieve high recognition performance. However, in recent years, instead of this model, methods using a DNN, which deepens the artificial neural networks, have been proposed [15–19, 24]. DNN-HMM has been proposed as one such model in which the GMMs of GMM-HMM are replaced with a DNN [15–19]. GMM-HMM solves the output distribution of the feature vector in each state of HMMs as a regression problem. On the other hand, DNN-HMM solves the posterior probability for each state of HMMs with respect to the feature vector as a classification problem.

DNN-HMM shows the effectiveness for modeling one-dimensional time series data, as described above. Therefore, it is expected that this model which expands DNN-HMM to two dimensions, can effectively model two-dimensional data such as images. By estimating the output probability corresponding to each state of SLHMMs using a DNN, DNN-HMM is extended to two dimensions. Our proposed method integrates SLHMMs and a DNN. Consequently, it can represent geometric normalization by discrete hidden variables included in the SLHMMs and estimate the output probabilities by the DNN.

3.1. Model structure

Figure 2 shows an overview of the proposed method. It consists of a DNN part that extracts features from image data and estimates the posterior probabilities over the SLHMM states, and an SLHMM part that captures the geometric structure of the target object by explicit modeling of the generative process. The proposed method has the state transitions probability and output probability as model parameters because the image is modeled by two Markov chains similar to SLHMMs. With the proposed method, the output probability is estimated by a DNN. Since a DNN models the relationship between image features and SLHMM states, the input is the image feature vector, and the output is the posterior probabilities of SLHMM states. Then, the output probability for each state in the SLHMMs is calculated from the posterior probability, which is the output of a

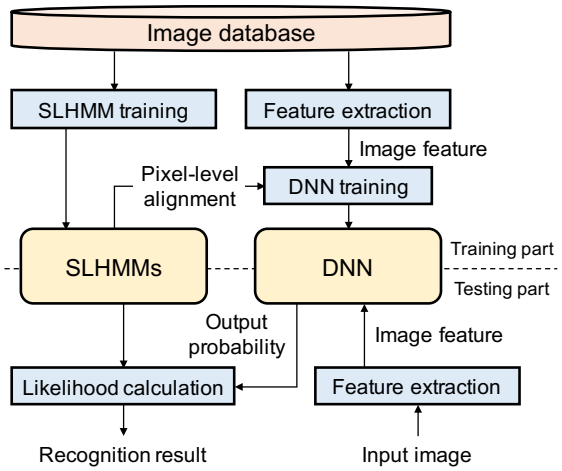

Fig. 2. Overview of proposed method

DNN; that is, the DNN calculates the posterior probability for all the SLHMM states of all classes. In the two-dimensional lattice $\bar{\boldsymbol{K}} = \prod_{n=1}^{N} \boldsymbol{K}_n$ including all the SLHMM states of all classes, the hidden variables are defined as $\bar{\boldsymbol{z}} \in \bar{\boldsymbol{K}}$, and the model parameter is defined as $\bar{\boldsymbol{\Lambda}} = \{\boldsymbol{\Lambda}_1, ..., \boldsymbol{\Lambda}_N\}$, where $n = (1, 2, ..., N)$ is a class number. Therefore, using this output probability, recognition based on the proposed method is performed. Flexible distribution representation based on a DNN enables highly accurate output probability estimation. As a result, it is expected that the proposed method can appropriately model an image, and the recognition performance improves upon that of conventional methods.

3.2. Training

In the training part of the proposed method, an SLHMM for each class is first trained from training data. Second, each pixel of training data is assigned an SLHMM state by forced state alignment and the correct state label is determined for each pixel using the assignment result. Forced state alignment, which is to assign the state sequence that obtains the maximum likelihood, is performed using the Viterbi algorithm:

$$\hat{\boldsymbol{z}}^{(m)} = \arg\max_{\boldsymbol{z}^{(m)}} P(\boldsymbol{O}, \boldsymbol{z}^{(m)} | \boldsymbol{\Lambda}), \tag{7}$$

where this operation is performed independently in two Markov chains. Then, a DNN is trained using the results of the forced state alignment as the correct state label. With the proposed method, the DNN takes image feature $\boldsymbol{O_t}$ as input and produces the posterior probability over all SLHMM states $P(\bar{\boldsymbol{z}}_{\boldsymbol{t}} | \boldsymbol{O_t}, \bar{\boldsymbol{\Lambda}})$ as output. Therefore, each output layer unit of the DNN corresponds to each SLHMM state. Moreover, a softmax function is used as the activation function of the output layer of the DNN to approximate the output as posterior probabilities. The DNN is discriminately fine-tuned using the back propagation method [25] using the cross entropy criterion. Although various image features can be used as input of the DNN, a pixel value vector, which collects several pixels surrounding each pixel, is input. The reason for using such an input is that it is raw data of an image and a feature amount in a local region.

3.3. Testing

In the testing part, feature extraction is first performed for each local region of input image $\boldsymbol{O}$ to obtain image feature $\boldsymbol{O_t}$. Second, the output probability $P(\boldsymbol{O_t} | \bar{\boldsymbol{z}}_{\boldsymbol{t}}, \bar{\boldsymbol{\Lambda}})$ of SLHMMs is calculated using Bayes' theorem with posterior probability $P(\bar{\boldsymbol{z}}_{\boldsymbol{t}} | \boldsymbol{O_t}, \bar{\boldsymbol{\Lambda}})$ obtained by inputting image feature $\boldsymbol{O_t}$ to the DNN. Bayes' theorem is written as

$$P(\boldsymbol{O_t} | \bar{\boldsymbol{z}}_{\boldsymbol{t}}, \bar{\boldsymbol{\Lambda}}) = \frac{P(\bar{\boldsymbol{z}}_{\boldsymbol{t}} | \boldsymbol{O_t}, \bar{\boldsymbol{\Lambda}}) P(\boldsymbol{O_t} | \bar{\boldsymbol{\Lambda}})}{P(\bar{\boldsymbol{z}}_{\boldsymbol{t}} | \bar{\boldsymbol{\Lambda}})}, \tag{8}$$

where $P(\bar{\boldsymbol{z}}_{\boldsymbol{t}} | \bar{\boldsymbol{\Lambda}})$ is the appearance probability for each state, and $P(\boldsymbol{O_t} | \bar{\boldsymbol{\Lambda}})$ is the occurrence probability of the image feature, and they can be ignored by assuming that they are uniform. Therefore, the proposed method performs recognition, as with Eq. (6), using the output probability estimated by the DNN.

4. EXPERIMENTS

Face recognition experiments on the XM2VTS database [26] were conducted to evaluate the effectiveness of the proposed method. We prepared 8 images of 100 subjects; 6 images were used for training, and 2 images were used for testing. Face images composed of 64×64 grayscale pixels were extracted from the original images. Figure 3 shows examples of face images in the XM2VTS database.

4.1. Comparison of conventional SLHMMs and DNN-SLHMM

In this section, the proposed method was evaluated by comparing it with conventional SLHMMs. SLHMMs and DNN-SLHMM with 16×16, 24×24, 32×32, 40×40, 48×48, 56×56, and 64×64 states were used. SLHMMs had a single Gaussian distribution for each state and used the pixel value for each pixel of the image as the input. In a DNN, a nine-dimensional pixel value vector obtained by taking pixels of eight neighborhoods for each pixel of the image was used as an input feature, and a one-hot vector based on the state alignment of SLHMMs was used as an output label. In these experiments, the DNN was a fully connected feed-forward neural network and had 1-hidden-layer. In the network architecture, there were nine units in the input layer and 1024 units in the hidden layer, and the number of units in the output layer was the total number of states included in SLHMMs for all classes. The sigmoid activation function was used for the hidden layer, and the softmax activation function was used for the output layer.

Figure 4 shows the results of the face recognition experiments. The recognition rate of DNN-SLHMM, exceeded those of SLHMMs, in all number of states. This improvement in recognition performance is due to advanced feature classification based on DNNs. The maximum recognition rate of 94.5% was obtained when DNN-SLHMM had the 48×48 states. DNN-SLHMM with 64×64 was significantly worse than that with 48×48. Under the condition with 64×64 states, since the number of states and the size of the input image were equal, the models had no ability to normalize geometric variations. Therefore, this result suggests that the geometric normalization based on the structure of the SLHMMs is effective for DNN-SLHMM.

4.2. Comparison with other methods

DNN-SLHMM was evaluated by comparing it with three SLHMM-based methods (**SLHMM**, **DCT-SLHMM**, and **GMM-SLHMM**) and two CNN-based methods [3, 4] (**CNN** and **CaffeNet**). The details of the SLHMM- and CNN-based methods are as follows:

Fig. 3. Examples images in dataset

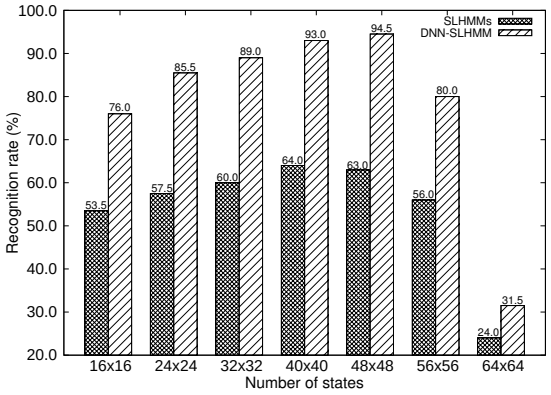

Fig. 4. Recognition rates of SLHMMs and DNN-SLHMM

SLHMM: The model structure was 32×32-state SLHMMs. A nine-dimensional pixel value vector was used for the input feature, as with the proposed method.

DCT-SLHMM: The model structure was 16×16-state SLHMMs. A 16-dimensional discrete cosine transform (DCT) coefficient value vector was used for the input feature. This input vector was obtained by taking the 4×4 of the low frequency region out of the DCT transformed 12×12 pixels of the surrounding pixels for each pixel of the input image.

GMM-SLHMM: The model structure was 16×16-state SLHMMs, and used GMMs for the output distribution. This method had two mixed Gaussian distributions for each state. In addition, a one-dimensional pixel value was used for the input feature.

DNN-SLHMM: The model structure was 48×48-state SLHMMs, using a DNN for the output distribution, and a nine-dimensional pixel value vector was used for input feature.

CNN: A CNN was trained using a Caffe [27]. The architecture of the CNN model was $I(64,1)$ – $C(128,10,1,55)$ – $P(3,2,27)$ – $C(256,5,1,23)$ – $P(3,2,11)$ – $F(800)$ – $F(600)$ – $F(400)$ – $O(100)$, where $I(i,d)$ indicates an input layer with a d dimensional $i \times i$ sized image, $C(f,w,s,o)$ indicates a convolutional layer with f filters of a $w \times w$ sized window, which a stride of s and $o \times o$ sized output, $P(w,s,o)$ indicates a pooling layer, $F(n)$ indicates a fully connected layer with n units, and $O(c)$ indicates an output layer with c classes. The rectified linear unit (ReLU) activation function was used in the convolutional and fully connected layers. The stochastic gradient descent (SGD) algorithm with a mini-batch of size 200 was used for training, and dropout with a probability of 0.5 was used for the convolutional and fully connected layers.

Table 1. Comparison of DNN-SLHMM with other methods

Method	Recognition rate (%)
SLHMM	62.5
DCT-SLHMM	92.0
GMM-SLHMM	73.5
DNN-SLHMM	94.5
CNN	82.5
CaffeNet	85.5

CaffeNet: A pre-trained CNN (CaffeNet) [27], which was trained using the ImageNet Large Scale Visual Recognition Challenge 2012 (ILSVRC2012) dataset [28], was used to extract image features. The image-feature vectors were composed of 4096 dimensions extracting the pre-trained CaffeNet of the 7th fully connected layer. The one-nearest neighbor was then used as the classifier.

Table 1 lists the experimental results of **SLHMM**, **DCT-SLHMM**, **GMM-SLHMM**, **DNN-SLHMM**, **CNN**, and **CaffeNet**. Comparing structures of the output distribution (**SLHMM**, **GMM-SLHMM**, and **DNN-SLHMM**), **DNN-SLHMM** achieved the highest recognition rate. This indicates that **DNN-SLHMM** overcame the lack of expression capability of HMM states. It also performed better than **DCT-SLHMM**, which used the heuristic feature.

DNN-SLHMM achieved better recognition rates than the CNN-based methods. This suggests that **DNN-SLHMM** is more effective than the CNN-based methods when the amount of training data is insufficient. However, the number of training images in the experiments was too small to train CNNs. Therefore, in the future, we should conduct comparative experiments on large datasets.

5. CONCLUSION

This paper proposed an image recognition method based on SLHMMs using a DNN for representing the output distributions. In face recognition experiments, the proposed method achieved high performance through comparison with conventional SLHMM-based methods. As a result, the proposed method is effective in image recognition. Future work will include an investigation of the architectures of the DNN in the proposed method, and an extension to the end-to-end model by replacing the SLHMMs in the proposed method with models based on neural networks.

6. REFERENCES

[1] M. Turk and A. Pentland, "Face Recognition Using Eigenfaces," *IEEE Computer Vision and Pattern Recognition*, pp. 586–591, 1991.

[2] S. Watanabe and N. Pakvasa, "Subspace Method of Pattern Recognition," *1st International Joint Conference on Pattern Recognition*, pp. 25–32, 1973.

[3] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Processings of the IEEE*, vol. 86, pp. 2278–2324, 1998.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Conference on Neural Infomation Processing Systems*, pp. 1097–1105, 2012.

[5] F. S. Samaria, *Face recognition using hidden Markov models*, Ph.D. thesis, University of Camgridge, 1994.

[6] A. V. Nefian and M. H. Hayes, "A Hidden Markov Model for Face Recognition," *International Conference on Acoustics, Speech and Signal Processing*, vol. 5, pp. 2721–2724, 1998.

[7] S. S. Kuo and O. E. Agazzi, "Keyword spotting in poorly printed documents using pseudo 2-D hidden Markov models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 842–848, 1994.

[8] A. V. Nefian and M. H. Hayes III, "Maximum likelihood training of the embedded HMM for face detection and recognition," *International Conference on Image Processing*, vol. 1, pp. 33–36, 2000.

[9] J. Li, A. Najmi, and R. M. Gra, "Image classification by a two dimensional hidden Markov model," *IEEE Transactions on Signal Processing*, vol. 48, pp. 517–533, 2000.

[10] H. Othman and T. Aboiilnasr, "A simplified second-order HMM with application to face recognition," *International Symposium on Circuits and Systems*, vol. 2, 2001.

[11] X. Ma, D. Schonfeld, and A. Khokhar, "Image segmentation and classification based on a 2D distributed hidden Markov model," *Society of Photo-optical Instrumentation Engineers*, vol. 6822, 2008.

[12] J. T. Chien and C. P. Liao, "Maximum confidence hidden Markov modeling for face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, 2008.

[13] D. Kurata, Y. Nankaku, K. Tokuda, T. Kitamura, and Z. Ghahramani, "Face Recognition based on Separable Lattice HMMs," *International Conference on Acoustics, Speech and Signal Processing*, vol. 5, pp. 737–740, May 2006.

[14] K. Sawada, A. Tamamori, K. Hashimoto, Y. Nankaku, and K. Tokuda, "A Bayesian approach to image recognition based on separable lattice Markov models," *IEICE TRANSACTIONS on Information and Systems*, vol. 99, no. 12, pp. 3119–3131, 2016.

[15] H. Bourlard and N. Morgan, *Connectionist Speech Recognition: A Hybrid Approach*, Kluwer Academic Publishers, Norwel, MA, USA, 1993.

[16] F. Seide, L. Gang, and Y. Dong, "Conversational Speech Transcription Using Context-Dependent Deep Neural Networks," *Interspeech*, pp. 437–440, 2011.

[17] G. Hinton, L. Deng, Y. Dong, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Neural Networks for Acoustic Modeling in Speech Recognition," *IEEE Signal Processing Magazine*, vol. 29, pp. 82–97, 2012.

[18] A. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic modeling using deepbelief networks," *IEEE Trans. Audio, Speech, & Language Proc.*, vol. 20, pp. 14–22, 2012.

[19] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large vocabulary speech recoginition," *IEEE Trans. Audio, Speech, & Language Proc*, vol. 20, pp. 30–42, 2012.

[20] B. H. Juang, S. Levinson, and M. Sondhi, "Maximum likelihood estimation for multivariate mixture observations of Markov chains," *IEEE Trans. Inform. Theory*, vol. 32, pp. 307–309, 1986.

[21] J. L. Gauvain and C. H. Lee, "Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains," *IEEE Trans. on Speech and Audio Proc*, vol. 2, pp. 291–298, 1994.

[22] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum Likelihood from Incomplete Data via The EM Algorithm," *Journal of The Royal Statistical Society (B)*, vol. 39, pp. 1–38, 1977.

[23] Z. Ghahramani and M. I. Jordan, "Factorical Hidden Markov Models," *Machine Learning*, vol. 29, pp. 245–273, 1997.

[24] N. Morgan, "Deep and wide: Multiple layers in automatic speech recognition," *IEEE Trans. Audio, Speech, & Language Proc.*, vol. 20, pp. 23–29, 2012.

[25] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, 1986.

[26] K. Messer, J. Mates, J. Kittler, J. Luettin, and G. Maitre, "XM2VTSDB: The Extended M2VTS Database," *Audio and Video-Based Biometric Person Authentication*, pp. 72–77, 1999.

[27] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 675–678.

[28] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.