International Journal of Innovative Research in Computer Science & Technology (IJIRCST)
ISSN: 2347-5552, Volume-6, Issue-3, May 2018
DOI: 10.21276/ijircst.2018.6.3.4

Improved HMM by Deep Learning for Ear Classification

Mohammed Ahmed Alhanjouri

***Abstract*— Ear recognition is one of the most relevant applications of image analysis. It's a true challenge to build an automated system which exceeds human ability to recognize ears. Humans do not identify the ears ordinarily, so we are not skilled when we must deal with a large number of unknown ears. The modern computers, with an almost limitless memory and computational speed, should overcome humans' limitations. This work uses Ear classification problem to improve Deviance Information Criterion- Structural Hidden Markov Model (DIC-SHMM) by Convolutional Neural Network (CNN). HMM is a strong model for large size of features. While the CNN, as a most important technology for deep learning, used for image classification, to recognize persons by their ears as a one of unique biometric physiological characteristics. Three systems will be used to classify ear images, deep learning for the original image directly, deep learning for eigenvector as Principle Components Analysis (PCA) of the original image to compare them with proposed combining convolution layers of CNN with improved HMM for the original Image. The proposed system shows the best correction rate as 97.5%.**

***Index Terms*—Improved SHMM, convolutional layers, Max-pooling, Ear Classification, Inverse Weighted Average K-means clustering, Deviance Information Criterion.**

I. INTRODUCTION

The first usage of HMM has been in speech/sounds recognition applications for few decades [1,2]. Later HMM are being applied to image recognition era. In 2000, the maximum likelihood training for the continuous mixture embedded HMM was presented and used for face detection and recognition [3], and SHMM was used for face recognition [4]. Since HMMs are one-dimensional in nature, many researchers have tried to represent the two dimensional structural. In (2002), a generalization of the embedded hidden Markov models was used for face recognition.

Manuscript Received April 28, 2018.
Mohammed Alhanjouri, Computer Engineering Department, Islamic University of Gaza, Gaza City, State of Palestine, (e-mail: mhanjouri@iugaza.edu.ps).

An application of the embedded Bayesian networks (EBNs) is presented for face recognition and introduced the improvement of this approach versus the "eigenface" and the embedded HMM approaches [5]. Later (2003), low-complexity 2D-HMM (LC 2D-HMM) was proposed, which consists of a rectangular constellation of states, where both vertical and horizontal transitions are supported. In (2004), another approach is the 1D discrete HMM (1D-DHMM), which models a face image using two standard HMMs, one for observations in the vertical direction and one for the horizontal direction [6]. On the other hand, biometric authentication that is uniquely identifying a person can be divided to two types, physiological or behavioral characteristics Physiological characteristics like as iris matching, fingerprint, hand geometry, palms, DNA, hand veins, face, ear, etc. Behavioral biometric can be a speech, gait, and keystroke and signature analysis. Ear recognition remains an unsolved problem and a demanded technology to support other bioinformatics features.[7] The input of the most ear recognition system is always an image or video stream. The output is verification of the person or an identification of some subjects that appear in the image or video [8]. Some approaches define the system in three main steps: ear detection (segmentation), feature extraction, and classification. In this work, convolutional layers in CNN will be used to extract the ear features from included and occluded faces. Then we introduced DIC-SHMM based K-means for classification as a strong model for these features, the generic system is shown in figure 1. To achieve our objective some pre-processing steps are needed, such as ear segmentation that is defined as the process of extracting ears from faces. So, the system positively identifies a certain image region as an ear. This step has many procedures like skin detection, and edge detection or curvature determination.

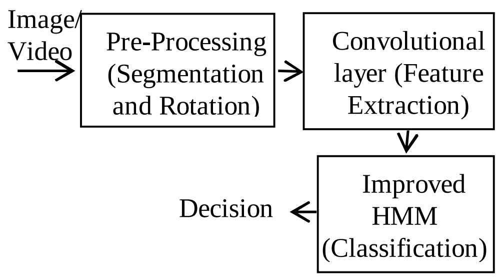


Figure 1: A generic proposed Ear recognition system

The next step is convolutional layers, which represents the feature extraction step in our models, involves obtaining relevant ear features from the data. And the last part is Improved HMM that recognizes the ear images. In an identification task (or recognition), the system would report an identity from a database. This step implements and trains improved HMM for each person, which use to evaluate posterior probabilities of unknown test sample to select the model with highest value.

II. PRE-PROCESSING STEPS

It is the first phase for any classification system; many steps can be achieved in this phase depending on the input images such as filtering, denoising, region extraction, rotation, segmentation resizing, etc. In this work, several preprocessing steps have been implemented to support the classification phase. Skin region extraction, era Image segmentation, rotation, gray level conversion, and image resizing are used to prepare the input ear images to be suitable for deep learning phase.

A. Skin Region Extraction

Firstly, the input RGB color input image from the AMI database is converted into YCbCr color image consisting of three components such as luminance, blue-difference (Cb) and red-difference (Cr). After converting into YCbCr color image, Cb, and Cr components are extracted from YCbCr color space to be pre-processing for segmentation of human skin.

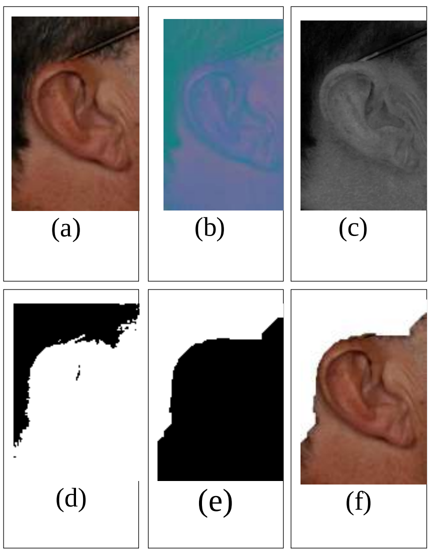


Figure 2: Skin Region extraction. (a) original image, (b) YCbCr color space image, (c) (Cb – Cr) components image, (d) after watershed transform and H/V filters, (e) after dilation morphological operation, (f) original skin region image.

This step is very important to separate skin and non-skin regions by computing the gradient magnitude, that achieved by applying horizontal and vertical filters [10]. The skin region will be extracted by applying the watershed transform and then dilation operation as shown in Figure 2.

Part (b) in figure 2 is YCbCr color space for image in part (a), it is important to calculate the deference between them (Cb – Cr) to produce part (c), then the watershed transform and Horizontal/Vertical filters have been applied to extract part (d), as shown it is not smooth enough to grantee good results for next steps, therefore dilation morphological operation has been used to get part (e) which applied as a mask on original image to extract the skin region in part (f).

B. Ear Image Segmentation

After extracting the skin region, we need to segment the ear boundaries by high pass filter to detect the ear edges. In this work, Canny filter has been used to detect the Ear boundaries for many reasons; Canny filter has good detection as it responds to edges, not noise, it has good localization as it detects edges near true boundaries, although it involves multiple stages: image smoothing, gradient estimation, non-maximum suppression, edge tracking. It is useful to extract continue and connected edges as possible [11]. The result of the Canny filter is shown in Figure 3 (a) that shows the edges in skin region only and removes others in the non-skin region. To extract ear image, erosion morphological operation has been applied to remove the small edges in skin region that appear on face around the ear as in part (b). From the upper and lower limits for horizontal and vertical axes, the segmented ear image has been extracted in part (c) as a gray level image.

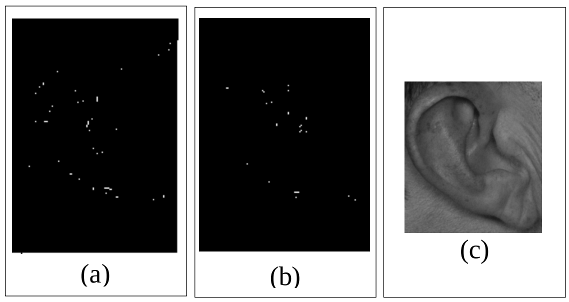


Figure 3: Ear image segmentation. (a) Canny filter result of Figure 2(f), (b) after erosion morphological operation, (C) exreacted original image corresponding to part (b).

C. Ear Image Rotation

After ear image has been segmented, we still need to another important pre-processing step which effects dramatically on classification stage. It is the rotation step to grantee that all ear images for the same person will be classified correctly while he moves his head up or down as illustrated in figure 4 that shows two pictures for the same ear; in part (a) he looks to up, while he looks down in part (b).

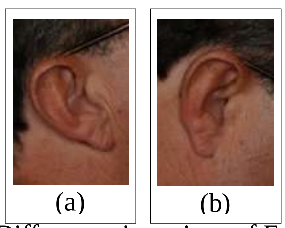


Figure 4: Different orientations of Ear images. (a) looking to up (b) looking to down.

International Journal of Innovative Research in Computer Science & Technology (IJIRCST)
ISSN: 2347-5552, Volume-6, Issue-3, May 2018
DOI: 10.21276/ijircst.2018.6.3.4

To achieve this important step, the dilation morphological operation has been applied to images in figure 3 (b) to produce figure 5 that is suitable to apply some geometry operation to calculate appropriate angle for rotation as shown in the following Algorithm:

Algorithm 1: automatic rotation of Ear Segmented Image

1- Extract the limit points of ear image, upper point (X^1), Lower point (X^2), Lift point (X^3), Right point (X^4)
2- Calculate the Mid-points horizontally (Mh) and vertically (Mv) as:
   $Mh = (x_1^4 + x_1^3) / 2$, and $Mv = (x_2^1 + x_2^2) / 2$.
3- Calculate the rotation angle as:
   $\theta = \sin^{-1} [(Mh - x_1^2) / (x_2^2 - Mv)]$,
4- Return to step 1 until the limit points unchanged.

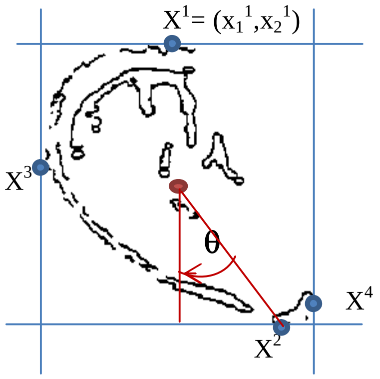


Figure 5: Era image rotation

According to this Algorithm, the ear images in figure 4 will be rotated as shown in figure 6, after convert it to gray level and crop the ear image at limit points and finally resize it (240 * 400) to be ready for Deep learning step that will extracting the feature and then classifying ears of 40 persons, 10 ear images for each one.

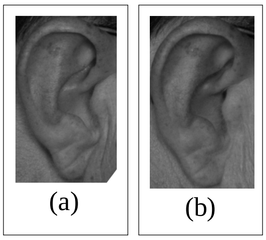


Figure 6: resultant image of rotation operation for parts (a) and (b) in figure 4, respectively.

III. Deep Learning

A deep Learning (deep neural network) is learning machine that uses artificial neural network that consists of multiple layers of hidden units to extract features of given inputs. A non-linear activation function is applied to each hidden unit to generate the output. The activation functions add nonlinearity to the model so that the deep neural network is able to model complex nonlinear relationships. The hidden units are connected with all or parts of the units in the previous layer and feed the outputs forward to the next layer. With the stacked structure of many hidden layers, the deep neural network is able to learn multiple levels of feature representations that correspond to different levels of abstraction. Analysis of the weights in each layer shows that the earlier layers can extract the lower level patterns from the inputs, and the latter layers tend to learn high-level features by combining the lower level patterns [12]. With such structures, the deep neural networks are able to extract complex representations. A deep neural network can be trained using standard backpropagation method such as described by Rumelhart et al. [1988] showed that it is possible to train the hidden units in a neural network to represent the important features of the task domain using backpropagation method. For each training input, the backpropagation algorithm first calculates the response of each unit in the network using the forward procedure. Next, the outputs are compared with the desired labels to calculate the error or loss. Then this error can be propagated back to all units and get the adjusted value of the unit parameters. The standard backpropagation method for training deep neural networks is the stochastic gradient descent (SGD) method. It converges much faster than the steepest descent method. In practice, an alternation of the stochastic gradient descent method is often used which is called “mini-batch” gradient descent. Other than calculating the gradient updates for each training sample, the “mini-batch” method uses multiple training samples at each step. This is a good compromise between the standard batch gradient descent and the global gradient descent method. One benefit is that with multiple training samples at each step, one can utilize the parallel computing tools to accelerate the training process. Another benefit is that using “mini-batch” makes the training process more stable than using individual inputs because it uses multiple training samples for gradient updates in each step.[13] Deep neural networks are a series of neural network structures that consist of many neural layers. With a deep structure, a neural network can effectively learn complicated mappings from the inputs to the target using end-to-end training, which does not rely on domain knowledge. Several techniques have been devised to overcome the difficulties in the training of complex neural networks [14].

A. *Convolution Neural Network*

A convolutional neural network (CNN) is a neural network that is suitable for processing images. The main part in CNN is the convolutional layer, which applies the "convolution" operation on the layer inputs to filter out the required information. The second important component is the pooling layer, which down-samples the inputs using a given selection method. The last is the fully connected layer, which combines all the outputs of the previous layer and generates the feature representations. With a deep structure, a CNN can effectively learn complicated mappings from raw images to the target, which requires less domain knowledge compared to handcrafted features and shallow learning frameworks. A typical CNN structure consists of several convolutional layers that have a small field of perception, and a few fully connected layers that each combines all the outputs from the last layer and yields a feature vector with several thousands of dimensions. To efficiently train the CNN model, the standard method is using Stochastic Gradient Descent (SGD) with backpropagation algorithm [15]. The CNN structure is originally designed for classification tasks, thus the last layer of the network works as a classifier based on the learned features from the immediate precedent layer. One advantage of these networks is that one can easily adapt a pre-trained neural network to new tasks where the only small number of training samples is available. This technique is called fine tuning. By removing the classifier layer of a neural network and attaching a new layer with randomly initialized parameters, one can train these new parameters efficiently and achieve good performance on the new task. Because the pre-trained model has already learned comprehensive representations through millions of training samples, the fine tuning procedure can start from exploring useful representations for the new task without going all the way from scratch [14].

Figure 7 shows CNN structure with (3×3) filter, the number of output channels corresponds to the number of filters (or kernels) in the filter bank for that particular layer.

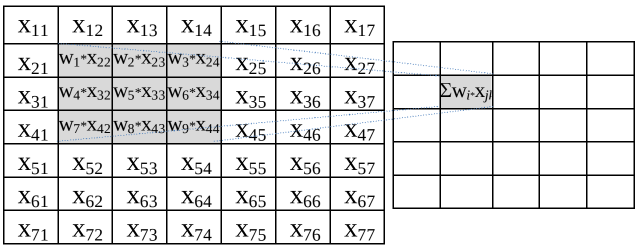


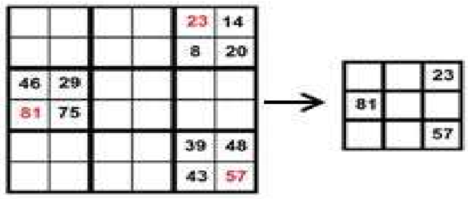
Figure 7: Convolution layer process

To convolve a filter (also known as the kernel) W with an input X we simply place the corresponding filter on the top left corner of our input and slide it across as shown in Figure 7. Every time we re-position our kernel we multiply the filter's weights by the corresponding values in the input tensor and reduce the output by adding up the result of each individual multiplication. If the spatial dimensions of the input are $n \times m$, we use a stride of 1×1 and our kernel's dimensions are $k \times k$ where k is odd and smaller than m and n then the spatial dimensions of the corresponding output are $n - (k - 1) \times m - (k - 1)$ as we don't allow the filter to be placed outside the image partially or otherwise; this mode of operation is referred to as "valid" mode. This mode is particularly helpful while pairing convolution with max-pooling and seeking deeper architectures as it allows us to perform an arbitrary number of convolutions without reducing the spatial dimensions of the current representation [14].

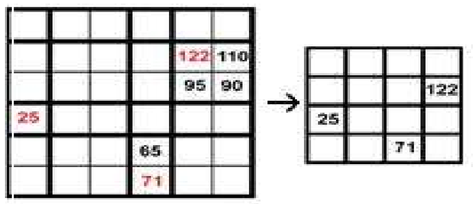
B. Max-Pooling

Max-Pooling is traditionally applied in combination with convolution and it is usually placed in between convolutional layers. We can think of this operation as placing a grid on top of the original input and pooling every element within the corresponding window as illustrated in Figure 8. Notice that every channel is processed independently (i.e. we only apply max-pooling across spatial dimensions) [16]. The max-pooling operation reduces the spatial dimensions of the input; reducing the size of the network as we move deeper into the architecture. More specifically, for an m × n input tensor, a $k \times k$ max-pooling operation, where k is smaller than m and n, also k, m, and n are all even, reduces the size of the representation down to $m / k \times n / k$. Max-Pooling endows the model with a degree of some translational invariance as the operation cares only for the magnitude of the activation and discards spatial information. Max-Pooling also increases the receptive field of neurons deeper in the network as their immediate input is affected by a larger and larger portion of the original input.

Max-Pooling and Fractional Max-Pooling are both used extensively throughout many researches. Unlike Max-Pooling however, Fractional Max-Pooling allows for varying window sizes and it is able to reduce the spatial dimensions of the input by a fractional amount and facilitate deeper architectures [17]. For example, under a fractional max-pooling window size of √2, the window grid may contain 2×2, 2×1, 1×2 and 1×1 pooling windows whose location follows a pseudorandom or random process every time a new sample is processed. The ability to sample a different pooling window configuration each time a sample is processed allows us to process the same sample multiple times and average the network output at test time; this technique is referred to as "model averaging" and it has been shown to produce even more accurate results [17].



(a) Max Pooling 2



(b) Fractional Max Pooling √2

Figure 8 Max-Pooling Types

An example of how a 6×6 input could be processed both by

International Journal of Innovative Research in Computer Science & Technology (IJIRCST)
ISSN: 2347-5552, Volume-6, Issue-3, May 2018
DOI: 10.21276/ijircst.2018.6.3.4

fractional max-pooling and regular max-pooling is shown in Figure 8 Notice how, under fractional max pooling, the spatial dimensions of the output are reduced to 4×4 while the spatial dimensions of the max pooling output are reduced down to 3×3. Notice as well that the arrangement of pooling windows under fractional max-pooling in this example is only one of many other possible ones.

C. Deep Neural Network Systems

In this work, the convolution layer with a total of 6 filters (3×3) is applied as denoted 6C3 in figure 9, this step will produces 6 images (248×398). Max-Spooling layer is applied to reduce the image size by Max-Pooling 3 to result 6 images (83×133). Another convolution layer is achieved to produce 12 images (79×129) by (5×5) filter as denoted 12C5, then Max-Pooling is applied to result 12 images (16×26) by 12S5. Finally, the full connection multi-layer perceptron is applied to act as a classifier for 40 persons with 4992 features for each one.

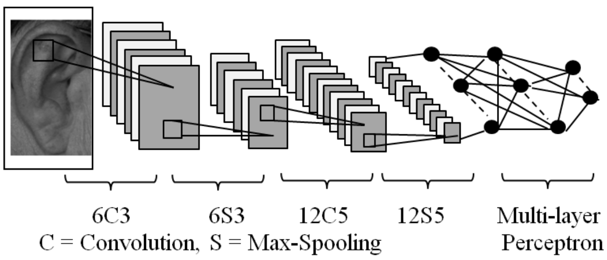


Figure 9 Deep Neural Network for Ear classification

This method takes a long time to accomplish the training phase, therefore, the PCA is used to reduce the dimensionality of input images. PCA depends on eigenvalues and eigenvectors of data, so it produces the curvature components in orthogonal directions. This is suitable for ear images as it contents many curves can be detected by PCA which achieved quickly. In this work, PCA is used to produce (69×112) ear image, that will be applied for the steps in Figure 9 by the same filters of convolution layers and the same type of Max-spooling to get 420 features for each person, these features feed the input neurons of full connection multi-layer perceptron. This approach dramatically reduces the time of training phase, as shown in the results section.

IV. PROPOSED IMPROVED SHMM

This work introduces another approach to reduce the training time. The deep neural networks consist of two main parts; feature extractor (few convolution and spooling layers), and classifier (Multi-layer perceptron). We will use the first part only to get the best features then use these features to feed Improved SHMM as a classifier.

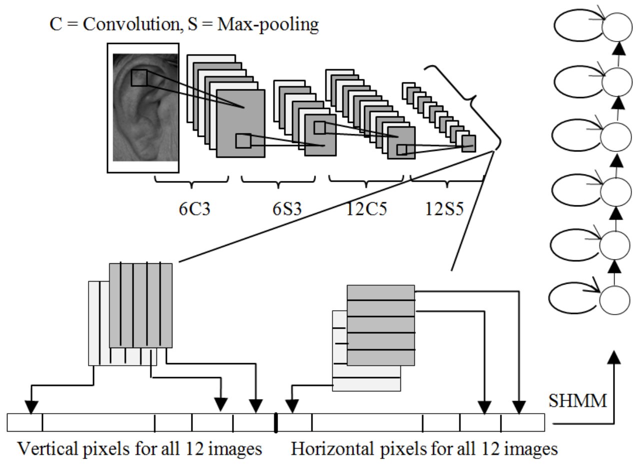


Figure 10 Proposed Improved SHMM System for Ear classification

SHMM is dynamic classifier because it can receive different sizes of inputs and depends mainly on some Gaussian mixtures that located in few states; in this work 6 states are used as shown in Figure 10. It uses one model for each person; therefore it is very fast in training phase. In the test phase, SHMM feed all models by the unknown features to select the model with highest posterior probability.

As in Figure 10, CNN feeds SHMM with 4992 horizontal features concatenated with 4992 vertical features to present 9984 features for each ear image. This step provides SHMM with good knowledge about changing rate horizontally and vertically.

In this work, we used SHMM which improved previously in [4] to overcome some problems in SHMM by using Deviance Information Criterion (DIC) as:

$$\mathrm{DIC} = \overline{\mathrm{D}(\mathrm{M})} + \mathrm{P_d} \qquad (1)$$

Where $\overline{\mathrm{D}(\mathrm{M})}$ is a measure of how well the model M fits the data which correspond to the expectation with respect to $\mathrm{P}(\mathrm{M}|\mathrm{X})$, $\mathrm{P_d}$ is the effective number of parameters of the model, and $\mathrm{D}(\mathrm{M})$ was defined as:

$$\mathrm{D}(\mathrm{M}) = -2\log \mathrm{P}(\mathrm{X}|\mathrm{M}) \qquad (2)$$

Where $\mathrm{X} = \{x_1, x_2, x_3 \ldots, x_c\}$. is the pattern contains sequence of local structure $\mathrm{S} = \{s_1, s_2, s_3 \ldots, s_c\}$. Let $\tilde{\mathrm{M}}$ is the posterior mean of effective parameter,

$$\mathrm{P_d} = \overline{\mathrm{D}(\mathrm{M})} - \mathrm{D}(\tilde{\mathrm{M}}) = \mathrm{E}(-2\log \mathrm{P}(\mathrm{X}|\mathrm{M})) + 2\log \mathrm{P}(\mathrm{X}|\tilde{\mathrm{M}}) \qquad (3)$$

The best model fits such data will have larger likelihood and smaller deviance. DIC was reformulated to,

$$\mathrm{DIC} = -2\log \mathrm{P}(\mathrm{X}|\tilde{\mathrm{M}}) + 2\mathrm{P_d} \qquad (4)$$

$\mathrm{P}(\mathrm{X}|\tilde{\mathrm{M}})$ can be obtained using forward algorithm for HMM. In order to calculate $\mathrm{P_d}$, $\mathrm{E}(-2\log(\mathrm{X}|\mathrm{M}))$ has to be approximated ,

$$\mathrm{P}(\mathrm{X}|\mathrm{M}) = \sum_{\mathrm{S}} \mathrm{P}(\mathrm{X},\mathrm{S}|\mathrm{M}) \approx \mathrm{P}(\mathrm{X},\mathrm{S}) \qquad (5)$$

$$\mathrm{P}(\mathrm{X},\mathrm{S}) = \mathrm{P}(\mathrm{S},\mathrm{X}) = \mathrm{P}(\mathrm{S}|\mathrm{X}) \times \mathrm{P}(\mathrm{X})$$

$$= P(s_c|s_{c-1} \dots s_2 s_1 x_c \dots x_1) \times$$
$$P(s_c s_{c-1} \dots s_2 s_1|x_c \dots x_1) \times P(X) \quad (6)$$
$$P(X|M) = \sum_{S}\left[\prod_{i=1}^{c} \frac{P(s_i|x_i)\,P(s_i|s_{i-1})}{P(s_i)} \times P(X)\right] \quad (7)$$

By substituting in Equation (3),

$$P_d = -2\sum_{S}\left[\prod_{i=1}^{c} \frac{P(s_i|x_i)\,P(s_i|s_{i-1})}{P(s_i)} \times P(X)\right] + 2\log P(X|\tilde{M}) \quad (8)$$

Still DIC-HMM has four problems: probability evaluation, statistical decoding, structural decoding, and parameter estimation.

The first problem is to determine the probability for such a model to produce such a sequence X. Equation (7) can be expressed as:

$$P(X|M) = \sum_{S}\left[\prod_{i=1}^{c} \frac{s_i(j) \times d_{i-1,i}}{P(s_i)}\right] \times \sum_{q} \pi_{q_1} b_{q_1}(o_1) a_{q_1 c_2} b_{q_2}(o_2) \dots a_{q_{(T-1)} q_T} b_{q_T}(o_T) \quad (9)$$

Then calculate P_d by substituting Equation (9) in Equation (3). The DIC value would be calculated by Equation (4). $P(X|\tilde{M})$ can be obtained by summing over forward probability in the forward algorithm as in traditional HMM.

The second problem is to determine the optimal state sequence. It is computed using Viterbi algorithm as in HMM.

The third problem is to determine the optimal structure sequence. It can be computed using Viterbi algorithm.

The last one is parameter estimation:

1. **D** was estimated by Baum-Welch optimization technique.
2. **S** was estimated by constrained Gaussian mexture posterior probability estimation technique.
3. **B** was estimated by DIC instead of using ML in HMM.
4. **A** and π were estimated as in traditional HMM.

Another weakness of HMM is that it depends on k-means clustering algorithm to defined Gaussian mixture for each model. K-means algorithm tries to partitioning a data set into K prototypes where these prototypes in some way best represent the data. K-means suffers of many problems; sensitivity to prototypes initialization, dead prototypes, converges to local optimum and needs specifying the number of clusters in advance. EM algorithm solves the last problem, but HMM model still suffer of the other ones. Actually, k-means uses Euclidian distance to centers the prototypes, first we need to normalize this distance to work will with classes have different number of data points. The average distance seems to be compromised choice, details is explained in [4].

V. Experimental Results

Our Algorithms is applied to Mathematical Analysis of Images (AMI) Ear Database (colored images) that is prepared and published by Universidad de Las Palmas [18].

All experiments are accomplished by HP i7 workstation desktop with specifications:

Memory 128 GB DDR4

Windows 10 Pro 64,

Intel® Xeon® E5-1660 v4 (3.2 GHz, 20 MB cache, 8 cores, Intel® vPro™)

Graphics card NVIDIA® Quadro® M4000 (8 GB)

HD 500GB SSD

To examine our systems and compare them, 400 Images from AMI database are used as 40 persons, 10 for each one. The data is divided into 40×7= 280 images for training, and 40×3=120 images for testing. In general, deep learning needs more data as it is designed for huge data. But the available databases for Ear images are limited and few images for each person, therefore this work compares between three different deep learning system by exhaustive time, and accuracy as shown in Table 1.

Table 1: Results of three Deep learning systems

	Accuracy	Training Time	Correct cases
Deep NN	94.2%	About 3 hours	113
PCA for Deep NN	91.2%	About 52 min.	110
Proposed SHMM	97.5%	About 18 min	117

The results show that the deep neural network without any modification takes a lot of time according to our system specifications, therefore the number of epochs is not enough to get very good results for 240×400 size images. While the deep NN after PCA (69×112) has less connection to reach the acceptable error in the training phase, but the images loss important features during reduction step. Proposed SHMM improving by DIC and IWAK algorithm shows the best accuracy and allows reducing the training time because there are no weights in full connections multi-layer perceptron. SHMM models each class individually, and for testing SHMM pass the features for all models the select which produces the highest posterior probability.

VI. Conclusions

This work discussed the pre-processing techniques for preparing the ear image for the classification task. The deep neural network is explained and roles of their layers. And in this work, three approaches using the deep neural network for ear image classification has been introduced, and the results illustrated that the combination of CNN and SHMM gave the best result of accuracy is (97.5%) by using twice convolution and spooling layers to extract features for improved SHMM classifier.

ISSN: 2347-5552, Volume-6, Issue-3, May 2018
DOI: 10.21276/ijircst.2018.6.3.4

References

[1] L.R. Rabiner, “A tutorial on hidden markov models and selected applications in speech recognition", IEEE Proc., vol. 77, no. 2, 1989, pp. 257–286.

[2] M. El-Hanjouri, W.Alkhaldi, N.Hamdy, O.Abdel Alim, “Heart Diseases Diagnosis Using HMM,” IEEE MELECON, pp. 489-491. 2002.

[3] A.V. Nefian, M.H. Hayes, “Maximum likelihood training of the embedded HMM for face detection and recognition”, Proc. Of the IEEE International Conference on Image Processing, ICIP 2000, Vol. 1, 10-13 September 2000, Vancouver, BC, Canada, pp. 33-36

[4] Mohammed Alhanjouri, Hana Hejazi, “DIC Structural HMM based IWAK-Means to Enclosed Face Data”, International Journal of Computer Applications, Volume 18, No.4, (0975–8887), March 2011

[5] Nefian A., "Embedded Bayesian networks for face recognition", Proc. of the IEEE International Conference on Multimedia and Expo, Vol. 2, , Lusanne, Switzerland, 26-29 August 2002, pp. 133-136, 2002.

[6] Le H., Li H., "Recognizing frontal face images using hidden Markov models with one training image per person", ICPR, vol. 1, p: 318–321, 2004.

[7] A. Pflug, C. Busch, A. Ross, “2D ear classification based on unsupervised clustering”, IEEE International Joint Conference on Biometrics (IJCB), 2014, FL, USA, 29 Sept.-2 Oct. 2014.

[8] B. Bhagabati, K. Sarma, “Application of Face Recognition Techniques in Video for Biometric Security: A Review of Basic Methods and Emerging Trends”, Chapter: 19, Handbook of Research on Modern Cryptographic Solutions for Computer and Cyber Security, 1st Edition, IGI Global, June 2016.

[9] Mohammed Alhanjouri, “Curvelet and Waveatom Transforms Based Feature Extraction for Face Detection", Alaqsa University Journal, Vol. 15, No.1.

[10] A. Das and D. Ghoshal, " Human Skin Region Segmentation Based on Chrominance Component using Modified Watershed Algorithm", Twelfth International Multi-Conference on Information Processing, Procedia Computer Science 89 (2016) 856 – 863

[11] B. Banchhor, T. Sharma, “Hybrid Approach for Face Detection Using Skin Color Based Segmentation and Edge Detection”, International Journal of Advanced Research in Computer and Communication Engineering, Vol. 3, Issue 6, June 2014.

[12] M. D. Zeiler, R. Fergus, "Visualizing and understanding convolutional Networks", In Computer Vision–ECCV Springer 2014, 818–833.

[13] L. Chen, "Pedestrian Detection Based on Deep Learning", Master Thesis in Electrical Engineering, University of California Riverside, September 2017.

[14] D. A. Mancevo, "Compressing Deep Convolutional Neural Networks", Master thesis in Machine Learning, KTH Royal Institute of Technology, June 2017.

[15] S. Basu, S. Mukhopadhyay, M. Karki, R. DiBiano, S. Ganguly, R. Nemani, S. Gayaka, "Deep neural networks for texture classification -A theoretical analysis", Journal of Neural Networks, Elsevier, January 2018

[16] P. Baldi, P. Sadowski, Z. Lu, "Learning in the machine: The symmetries of the deep learning channel", Journal of Neural Networks, Elsevier, November 2017

[17] G. Benjamin, “Fractional max-pooling." arXiv preprint arXiv:1412.6071 (2014).

[18] E. Gonzalez, L. Alvarez, L. Mazorra, “AMI EAR DATABASE”, Universidad De Las Palmas De G.C., http://www.ctim.es/research_works/ami_ear_database/#cita

Moahmmed Ahmed Alhanjouri received Bachelor of Electrical and Communications Eng. (honor) (1998), Master of Electronics Eng. (excellent) (2002), and Ph.D. in 2006. He has worked at Arab Academy for Science & Technology & Maritime Transport (Alex., Egypt) from 2002 to 2006. Then he worked at University of Palestine as dean of library, then as head of Software Eng. Dept. and he worked in Islamic university of Gaa as Assistant Professor, with responsibility: director of projects and research center, and head of computer Engineering department at Islamic University of Gaza, Palestine. currently he is serving as Associate Professor with responsibilities: Dean of Quality and Development, Chairman of IT Affairs, and Director of Excellence and eLearning Center at Islamic University of Gaza.
He has more than 40 scientific research papers in fields: AI, DSP, digital control, big data, and Biomedical Eng.

