Hybrid Hidden Markov Models and Genetic Algorithm for Robust Automatic Visual Speech Recognition

Amina Makhlouf[1], Lilia Lazli[1], Bachir Bensaker[2]
[1]Laboratory of Computer Research (LRI)
Department of Computer Science
University of Badji Mokhtar
[2]Department of Electronics
[1, 2]Faculty of Science of the Engineering
P.O. Box No. 12, El Hadjar 23200, Annaba, Algeria
{makhlouf_amina, l_lazli, bensaker_bachir}@yahoo.fr

ABSTRACT: *In this paper, we investigate the performance of an Automatic visual speech recognition system (AVSR) using the Hidden Markov Model (HMM) which is optimized by a Genetic Algorithm (GA). The search through the space of HMM structures is combined with optimization of emission and transition probabilities using the Baum-Welch algorithm in the training process. Compared with a HMM-based system, we observe a significant 13-31% improvement of the recognition rates.*

Keywords: Genetic Algorithm, Hidden Markov Model, Hybrid Model, Automatic Visual Speech Recognition

Received: 21 May 2013, Revised 27 June 2013, Accepted 30 June 2013

1. Introduction

Lip reading is hardly a new form of communication: it is a primary tool of the deaf community, not to mention a useful skill for transmitting messages across a loud party or over the heads of children. A new form of technology, however, is attempting to mechanize this age-old method of communication. Automatic lip reading, also known as automatic speech reading, is a growing branch of speech recognition technology. In theory, by monitoring a speaker's lip movements and other related elements, specially designed computer programs can interpret verbal messages even when noise interference or other obstacles prohibit a human voice from being properly heard.

Usually, automatic lip reading requires a video of the speaker. The shapes and rhythms formed by the lips are often the most important factors in interpretation, but a variety of other movements can help to decipher the speaker's message, as well. Facial expressions and movements of the head can be determining factors.

The development of automatic lip reading technology rests on the ability to identify isolated words that use the same lip movements every time they are spoken. Lip Geometry Estimation (LGE) is one of the more advanced computer systems in place that monitors and interprets such patterns.

In particular, a large number of studies have demonstrated that we could extract a lot of information from the mouth [1], [2], [3].

The first attempt to use vision to aid speech recognition systems was done in 1984 [1]. In his work the author demonstrated that visual speech yields information that is not always present in the acoustic signal and it enables improved recognition accuracy over purely acoustic-based speech recognition systems. In [2] authors developed a visual only speaker identification system using only the lip contour information. To this end shape deformations of the lip contours were modeled temporally by a HMM classifier. By utilizing the lip contour, such as height, width and area, a speech recognition system was developed with real-time tracking by using multi-stream HMM without automatic weight optimization [4].

The work in [5], concentrates on the visual front end for hidden Markov model based automatic lip-reading. Two approaches for extracting features relevant to lip-reading, given image sequences of the speaker's mouth region, are considered: A lip contour based feature approach, which first obtains estimates of the speakers lip contours and subsequently extracts features from them, and an image transform based approach, which obtains a compressed representation of the image pixel values that contain the speaker's mouth.

HMM are probabilistic finite state machines used to find structures in sequential data. An HMM is defined by the set of states, the transition probabilities between states, and a table of emission probabilities associated with each state for all possible symbols that occur in the sequence. They have many applications in system and process modeling, signal processing, pattern recognition and speech recognition [3], [5], [6]. HMM is considered as a basic component in speech recognition systems. Optimal model parameters estimation improves the performance of the recognition process.

In this paper, we aim to find the optimal candidate structure of HMM by using GA technique. The main point of this work is based on the quality of data modeling (observations) by performing HMM. Our goal is to propose a combination of algorithms that improve this quality. The criterion used to quantify the quality of HMM is the probability that a given model generates a given observation. Experimental results showed that our GA for HMM training can achieve more optimized HMM structure than the Baum-Welch algorithm.

The rest of this paper is organized as follow. In the next section we give background knowledge to understand our proposed approach, and the method used in this work. In Section III, the performance of all the system is evaluated and discussed. We conclude in section IV.

2. System Description

The visual information relevant to speech is mostly contained in the motion of visible articulators such as lips, tongue and jaw. In order to extract this information from a sequence of video frames it is advantageous to track the complete motion of the speaker s face and facial features.

We first introduce the concept of illumination compensation, where we try to reduce the dependency of light from over- or under-exposed images. As a precursor to lip segmentation, one of the detection techniques used was the object detection algorithm proposed by Viola-Jones [7]. This algorithm has been experimented with different color spaces that have reached interesting conclusions. Following face detection we implemented a mouth localization phase. We complete the visual front end by a feature extraction step which is based on the Discrete Cosine Transform (DCT). The visual recognition unit uses the stochastic approach based on the HMM hybridized with GA in the learning process as shown in Figure 1.

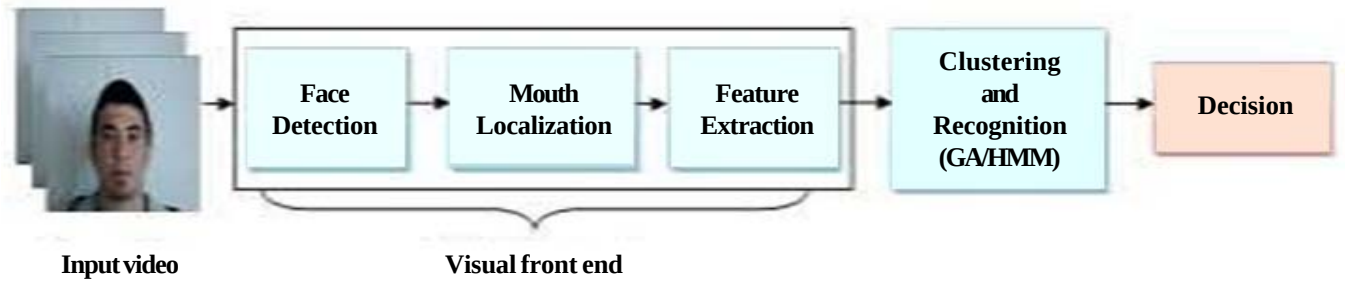


Figure 1. Automatic visual speech recognition system

Figure 1 shows the different stages in the learning and recognition processes of the proposed system. Each element in this figure will be described in the following.

2.1 Visual Front End

2.1.1 Face Detection and Mouth Localization

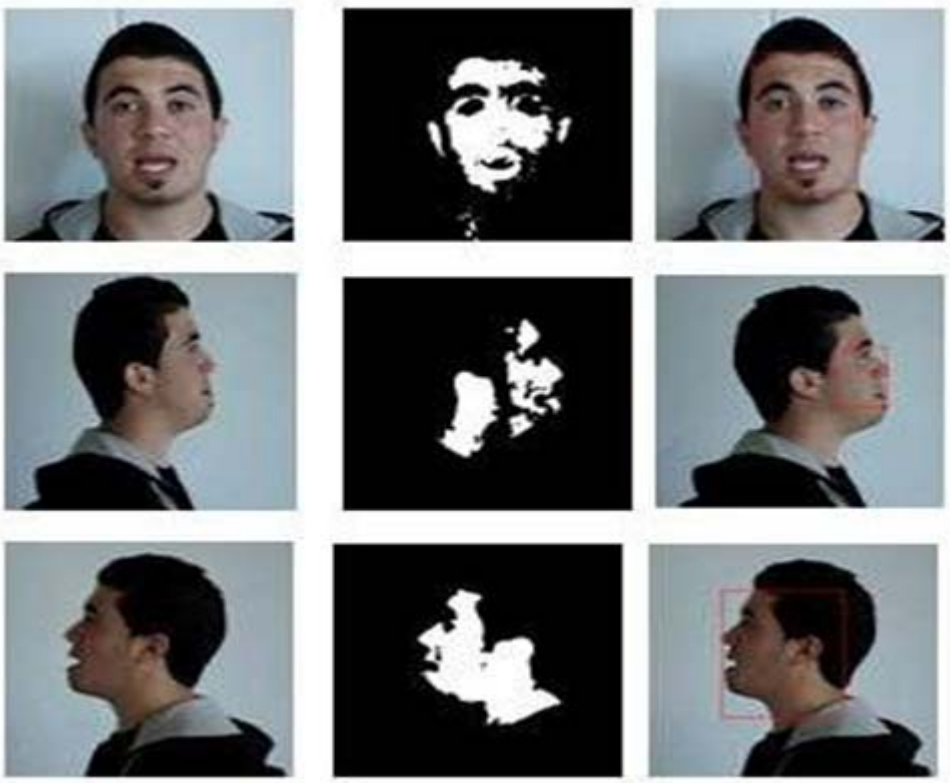
In the face detection step, Viola and Jones face detector algorithm is used [7]. The face detector is then applied over the detected skin regions to detect the target face, and a rectangular region is constructed with the same width as the detected face with a height of four times longer than the length of the face of the target model as shown in Fig. 2. In brief, the face detector consists of three main parts:

a) The integral image: representation of the image which allows calculating Haar-like features very quickly. In every image, a rectangular area can be defined and the sum of the values of its pixels calculated. A Haar feature is a simple linear combination of the obtained sums.

b) Adaboost classification of Haar-like features obtained rom Haar basis functions or Haar filters. Here, Adaboost is used both in selection and training of features. It combines a set of weak classifiers to form a stronger classifier [8].

c) Combination of classifiers in a cascade structure to increase the detection speed. In each step of a cascade structure, the sub-windows are reduced. Indeed, a large number of false negatives are removed by adjusting the classifier thresholds with small processing time. This structure results in keeping only strong classifiers for the final face detection.

A typical human face follows a set of anthropometric standards which have been utilized to narrow the search of a particular facial feature to smaller regions of the face [10]. It is possible to extract areas from face's geometry where the points should be



(a) (b) (c)

Figure 2. An example of face detection: (a) original image (b) skin detection with noise suppression (c) facedetection result

Detecting the middle of the mouth is not as simple as it is thought. To this end there are a lot of possibilities such as going over gradient horizontal and/or vertical decent, hue or saturation [9]. The possibility of the use of distinct hue of the lips taking into account the reflected light is implemented. This point is fetched by a defined hue value. In contrast to other methods, this method is not light independent, thus intensity and direction of the light can influence results.

In our work we use the following generic steps for the facial feature detection and extraction from the localized face image as shown in Figure 3:

a) For color image, convert it into gray scale image and adjust the intensity of both the type of images.

b) Find the gradient of the region of interest (ROI) in detected image using Sobel/Prewitt edge detection operator and then take lower part of face and project it vertically to get mouth localization.

c) Draw rectangular box on each of the detected feature components.

Figure 3. Examples of extracted lip ROI from our Arabic audio-visual database

2.1.2 Visual Feature Extraction

Once the ROI is isolated, it is recommended to extract useful information by using a minimum number of attributes to avoid statistical modeling difficulties due to high dimension of space attributes.

To characterize the video signals, in this work, we use the DCT. The main advantage of the DCT is its high compaction of the energy of the signal into a few DCT coefficients and the availability of a fast implementation of the transform, similar to the Fast Fourier Transform (FFT) [18]. The DCT method is widely used in many areas as image compression and feature extraction.

In this paper, the DCT is used to minimize the number of bits required to represent the information in an image. Thus removes the redundancy between neighboring pixel values. The DCT transforms each color component into DCT coefficients by using following relation [11]:

$$F(u,v) = \frac{1}{\sqrt{MN}} \alpha(u)\,\alpha(v) \sum_{x=1}^{M} \sum_{y=1}^{N} f(x,y) \cos\left[\frac{2(x+1)u\pi}{2M}\right] \cos\left[\frac{(2y+1)v\pi}{2N}\right] \quad (1)$$

Where M and N are the dimensions of the image, u is the horizontal spatial frequency, v is the vertical spatial frequency, $f(x, y)$ is the pixel value at coordinates (x, y), $F(u,v)$ is the DCT coefficient of size $M \times N$ at coordinates(u, v) and $\alpha(\bullet)$ is defined as follows:

$$\alpha(w) = \begin{cases} \frac{1}{\sqrt{2}} & w = 1 \\ 1 & otherwise \end{cases} \quad (2)$$

For images, the 2-dimensional DCT is used with the most visually significant information being concentrated in the first few DCT coefficients ordered in a zigzag pattern from the top left corner as shown in Figure 4.

Information concentration in the top left corner is due to he correlation between image and DCT properties. The visual DCT coefficients are subsequently extracted by fitting the DCT to each frame in the video. In order to limit the memory size and the necessary calculations for the learning and the recognition we kept only the first 100 coefficients to represent the image [12].

2.2 Vector Quantization

Vector quantization is used in many applications such as image and voice compression, voice recognition (in general statistical pattern recognition), and surprisingly enough in volume rendering. It is used to analyze a distribution and to describe it in a simplified representation.

The *K*-means algorithm is the most known algorithm for vector quantization. Its main idea is to randomly select a set of a priori fixed centers and iteratively find the optimal partition. Each observation is assigned to the nearest center (also called centroïd). After that the assignment of all data the average of each cluster is calculated. Thus constitutes to the new cluster representative. When it reaches a steady state (no data exchange the cluster) the algorithm is stopped. Figure 5 gives an example of application of the *K*-means algorithm.

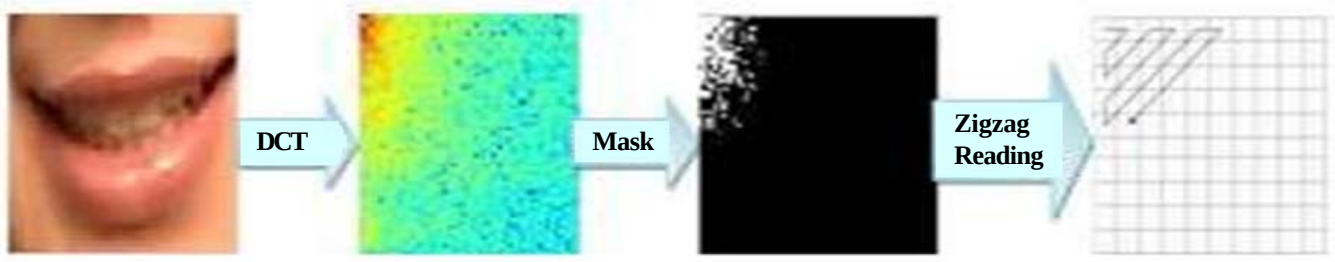


Figure 4. Process of choosing the DCT coefficient

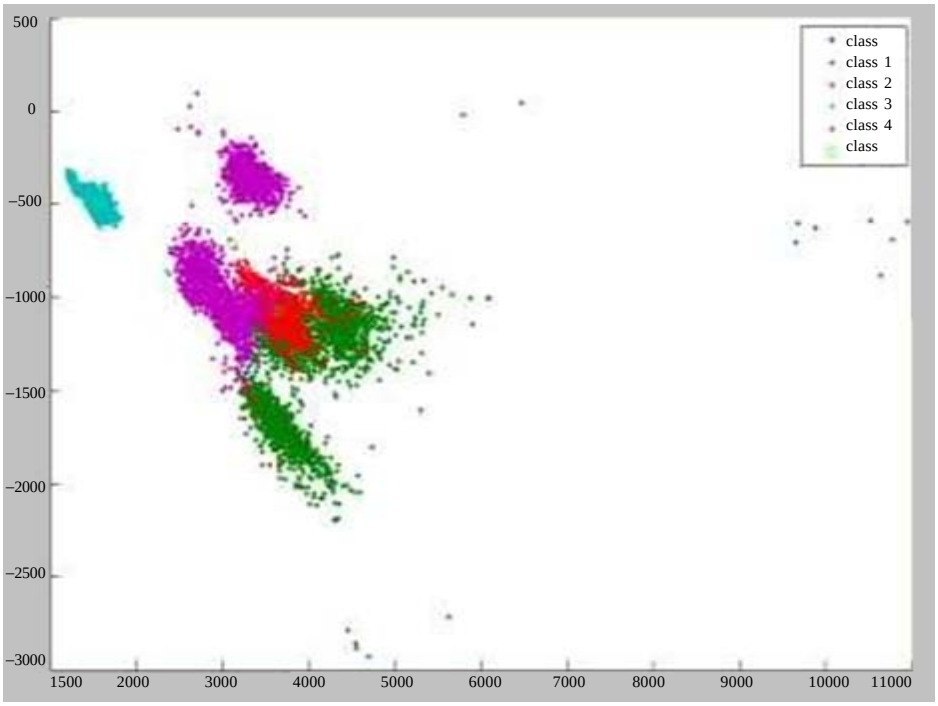


Figure 5. 2D representation of the visual database with 4 clusters

2.3 GA/HMM Recognizer

Generally, it is agreed that HMM is a useful statistical modeling tool for information extraction. But there are two outstanding issues. Firstly, how to determine the topology of the HMM and secondly, what is the optimized model parameters for accurate representation of the training data?

A *GA* is a robust general purpose optimization technique which evolves a population of solutions [20]. It is easy to hybridize other algorithms such as Baum-Welch training within a *GA*, in order to overcome the disadvantages signaled using these algorithms.

Furthermore, it is possible to design operators which favor biologically plausible changes to the structure of an HMM.

2.3.1 Learning

HMM is a class of models in which the distribution that generates an observation depends on the state of an underlying but unobserved Markov process [19]. Thus an HMM is a combination of two processes, namely a Markov chain which determines the state at time t, $S_t = s_t$, and a state-dependent process which generates the observation $X_t = x_t$ depending on the current state st. In most cases a different distribution is imposed for each possible state of the state space. A hidden Markov Model is characterized by:

- S: the set of states
- O: the sequence of observations, each of which is drawn from a vocabulary, $<o_1, o_2, \ldots, o_T>$
- N is the number of states in the model.
- M is the number of mixtures in the random function.
- Π: initial state probabilities
- A: the transitional probability matrix
- B: the emission probabilities, where $b_i(o_t)$ is the probability of observation ot generated from state i

In this paper, we explore the use of GAs for evolving HMMs in the training stage [13], [14], [15] and [16]. The GA will manipulate individuals who are going to be candidate solution to the problem that we want to resolve. Naturally, in the problem of learning of the HMM, the individuals are HMM. We must now encode the HMM in chromosomes on which will apply the genetic operators.

We give below the frame of this algorithm which will seek to obtain optimal HMM. We use a marking, named "*parent*" simply to treat only the necessary individuals during the optimization and the evaluation steps.

1- Initialization: Create randomly a population of size S. The most natural encoding is to fabricate the chromosome by reorganizing all the coefficients of the HMM. The simplest way is to juxtapose all rows of all matrices. We so obtain a coding in real numbers while respecting the constraints related to the HMM. The representation of a population is defined as follows:

π_1	…	π_N	$a_{1,1}$	…	$a_{1,N}$	$a_{2,1}$	…	$a_{N,N}$
$b_{1,1}$	…	$b_{1,M}$	$b_{2,1}$	…	$b_{N,M}$			

Figure 6. Chromosome representation in the GA/HMM training

No individual is marked "*parent*". Read an observation O

2- Optimization: Apply to each HMM of the unmarked population "*parent*" the Baum-Welch algorithm from the observation O.

3- Evaluation: The quality of an individual (also called fitness) describes the adequacy of this one with its environment. More precisely the individual will have a note as high as it will be a good solution to the problem. In the problem of optimizing an HMM, it is desired to quantify the ability of an HMM to learning an observation. In our GA, the fitness values are the results of the objective function. To this end the likelihood probability, $P(o_j|\lambda_i)$, is an appropriate criterion as objective function to determine the quality of the chromosomes.

The probability $P(o \mid \lambda)$ is calculated by the implementation of maximum likelihood method or Baum-Welch as optimization algorithm. The two algorithms have to maximize the probability such that a given HMM λ_i generated the training utterances o_j as follows:

$$f(\lambda_i) = \frac{P_n}{\sum_{i=1}^{N} P_i} \tag{3}$$

$$P_n = \frac{\sum_{i=1}^{M} \log\left(P(o_i|\lambda_i)\right)}{M} \tag{4}$$

It is proven that Baum-Welch algorithm leads to a local maximum of function $f(\lambda_i)$. Thus, it is possible that other better maxima

of $f(\lambda_i)$exist for given training set. In this paper we tried to overcome this problem by using genetic algorithms for maximization of $f(\lambda_i)$ as in [13].

4- Selection: Among all the individuals of the population, select a number $S' < S$, which will be used as parents to regenerate the S-S other individuals not selected. The selection is done according to the best calculated scores in step 3. Each selected individual is marked "*parent*".

5- Crossover/recombination: For each unmarked individual "*parent*" randomly select two individuals from the population of those marked "*parent*" and the cross. The crossover is in a crossover point, and is realized between two rows of the matrices of the HMM. This allows obtaining in return two children. It retains randomly only one of both children.

6- Mutation/normalization: On each unmarked individual "*parent*" we apply the mutation operator. This one consists in modifying a small random quantity each coefficient of the matrices of the HMM. Each coefficient is modified according to the value of the mutation probability. After treating an individual, we apply on him an operator of normalization to ensure that this individual still answers the constraints of the HMM. We should verify that the matrices of the HMM are stochastic. This operator is applied after the operation of mutation because subsequent operations imperatively work on HMM.

7- Evaluation of the stop condition: If the maximum number of iterations is not reached, then return to step 2, otherwise go to step 8.

8- Finally return the best HMM among the current population.

Note that this algorithm has been adapted to optimization of observation vectors. The re-estimation made so that the HMM has a maximal probability to generate the set of vectors.

2.3.2 Recognition

Recognition is done by a discriminant model. That is to say that learning will be associated with each word learned an HMM. Recognition will be done by calculating for each known HMM its probability of generating he word to recognize. We will recognize the word which the associated HMM obtained a maximum score.

Decision stage chooses the HMM who has the highest probability of generating the input data. In our work the decision is performed using Viterbi algorithm.

The idea of the Viterbi algorithm is to find the most probable path for each intermediate and finally for the terminating state in the trellis. At each time t only the most likely path leading to each state s_i '*survives*'.

3. Experimental Results And Discussion

3.1 Database

In this first work, a multi-speakers database was built, this database was recorded in a real environment (classroom very noisy), and it contains pronunciations of Arabic words isolated.

Video data were captured with a 690 × 340 pixel resolution at 30 frames/s frame rate and saved in AVI format. The database is collected from 11 speakers with variations of pose (profile and frontal view), these speakers are from different regional dialects, and each speaker pronounces each word 7 times (Arabic numerals from zero to nine) with different modes of pronunciation (normal, slow and fast). The distance between the camera and the speaker as well the luminance are adjustable to add diversification in the visual stream during the learning, the average distance is to 14.5 cm. In our basic corpus which contains only isolated words, the size of each record is 2 seconds which is enough time to utter a word slowly in Arabic.

3.2 Discussion of The Obtained Results

In our work we applied the proposed ASR system on our Arabic audio visual database, to achieve this recognition system; we have used 1/3 of the data for the learning stage and the remaining 2/3 to test the effectiveness of our system.

An ASR system using DCT coefficients as visual features, and a GA/HMM for the visual speech modeling, was implemented as

described in the previous sections. The visual feature extraction (DCT) is implemented in matlab, and The GA/HMM recognizer was built using the HTK (Hidden Markov Model Toolkit) [17].

In order to evaluate performance, various kinds of instance with different GA control parameters have been solved with our algorithm. We ran each instance 15 times with a different number of clusters, crossover probability values between 0.5- 0.9, and mutation probability with only the value 0.01, and obtained the maximum P(o | λ) values after 50 generation and the best performances was listed in Table 1 as follows:

Figure 7. Some examples of frames of our Arabic database

Number of clusters	P_c	P_m	Average P(o | λ)
3	0.5	0.01	–7.7629
4	0.5	0.01	–7.0046
7	0.8	0.01	–7.1555
9	0.8	0.01	–7.6595
12	0.9	0.01	–7.8234

Table 1. GA Parameters Training HMM for the Visual Database

We observe that the results are varied according to the parameters training of the *GA*. The number of clusters obtained by the vector quantization phase is 4 clusters with $P_c = 0.5$ and $P_m = 0.01$. This result is superior to all the other cases as one can see in the table1. Therefore, we use theme in our GA/HMM system.

Figure 8 gives the rate of recognition with respect to the clusters used in the experiment.

Based on Figure 8, we can see that the recognition rates obtained with our proposed GA/HMM algorithm with 4 clusters for clustering are better compared to those obtained with the standard HMM system, and the percentage increase between 13-31%.

This increase in performance is due to the ability of *GA* in generating and optimizing the model for each class. Optimizing performance of HMM systems using *GA* can be done by observing the parameters of *GA* to the system then test one by one

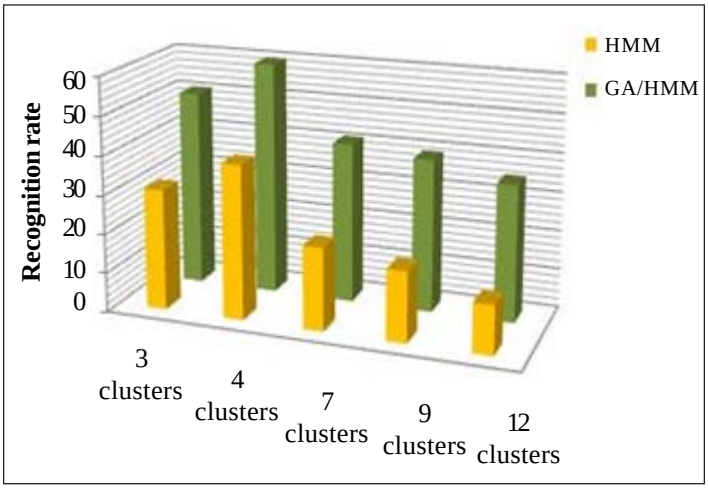


Figure 8. Comparison of recognition rate between HMM and GA/HMM on AVSR system

these combinations. However, using GA has the ability to optimize, but sometimes also can produce solutions that are not optimal.

4. Conclusion and Perspectives

This primary work propose an AVSR system that uses DCT coefficients as visual features and a GA to optimize the HMM structure for modeling the visual speech.

The AVSR system is tested using our own Arabic database. Based on the experimental results, we conclude that the system modeled by HMM and trained by our GA/HMM training produces a better performance than using the HMM trained by the Baum-Welch algorithm.

We are planning to cover more issues about improving the performance of the ASR system as a future work. We will try to enhance the size of our database and increase the number of speakers; also we will try to use another type of HMM in order to optimize them by the GA. Finally, we will also work on the comparison of our system with other systems in the literature by using one of standard databases.

References

[1] Petajan, E. (1984). Automatic lipreading to enhance speech recognition. *Global Telecommunications Conference*. p. 265-272.

[2] Luettin, J., Thacker, N. A., Beet, S. W. (1996). Speaker identification by lipreading, *In*: Proceedings of the 4th International Conference on Spoken Language Processing (ICSLP'96).

[3] Rabiner, L., Juang, B, -H. (1993). Fundamentals of Speech Recognition. Englewood Cliffs, *NJ: Prentice Hall*.

[4] Chen, T. (2001). Audiovisual speech processing, *IEEE Signal Processing Magazine*, 18 (1) 9-21.

[5] Adjoudani, A., Benoit, C. (1996). On the integration of auditory and visual parameters in a HMM-based asr, *In*: David Stork and Marcus Hennecke, editors, *NATO ASI: Speechreading by Humans and Machines*. SpringerVerlag.

[6] Potamianos, G., Graf, H. P., Cosatto, E. (1998). An image transform approach for HMM based automatic lipreading, *In*: Proceedings of the International Conference on Image Processing, 3, p. 173–177.

[7] Viola, P. A., Jones, M. J. (2001). Rapid Object Detection using a Boosted Cascade of Simple Features. *CVPR* , 1, p. 511-518.

[8] Papageorgiou, Oren, M., Poggio, T. (1998), A General Framework for Object Detection, *In*: International Conference on Computer Vision ICCV'98.

[9] Pai, Y., Ruan, S., Shie, M., Liu, Y. (2006). A Simple and Accurate Color Face Detection Algorithm in Complex Background. *In*: ICME, p. 1545-1548.

[10] Khandait, S. P., Khandait, P. D., Dr. Thool, P. D. (2009). An Efficient Approach to Facial Feature Detection for Expression Recognition, *International Journal of Recent Trends in Engineering*, 2 (1).

[11] ISIDOROS, R. (2008). Feature Extraction Optimization and Stream Weight Estimation in Audio-Visual Speech Recognition, a Phd thesis from Technical University of Crete.

[12] Neti, C., Potamianos, Luettin, J. (2000). Audio-visual speech recognition. Final Workshop 2000 Report, Center for Language and Speech Processing, *The Johns Hopkins University, Baltimore, MD*, Oct. 12.

[13] Goh, J, Tang, L., Al turk, L. (2010). Evolving the Structure of Hidden Markov Models for Micro aneurysms Detection, *Computational Intelligence* (UKCI), UK Workshop on, p. 1-6.

[14] Xueying, Z., Yiping, W., Zhefeng, Z. (2007). A Hybrid Speech Recognition Training Method for HMM Based on Genetic Algorithm and Baum Welch Algorithm. *IEEE Transactions on Neural Network*.

[15] Oudelha, M., Ainon, R. N. (2010). HMM parameters estimation using hybrid Baum-Welch genetic algorithm, *In*: (ITSim): p. 542–545.

[16] Slimane, M., Venturini, G., Asselin de Beauville, J. P., Brouard, T., Brandeau, A. (1996). Optimizing HMM with a genetic algorithm, Artificial Evolution, *Lecture Notes in Computer Science*, Springer Verlag, 1063, p.384-396.

[17] Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X. A., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., Woodland, V. *The HTK Book* (for HTK Version 3.4).

[18] Ahmed, N., Datta, S., Mulvaney, D., Farooq, O. (2008). A Comparison of Visual Features for Audio-Visual Automatic Speech Recognition, Acoustics'08, June 29- July 4.

[19] Ephraim, Y., Merhav, N. (2002). Hidden Markov processes, *IEEE Trans. Inform. Theory*, 48, p. 1518-1569, June.

[20] David E. Goldberg. (1989). Genetic Algorithms in Search Optimization and Machine Learning. Addison Wesley, p. 41.