

Comparative study of Gesture Recognition Using Gaussian Mixture Model, Support Vector Machine

Ms. S.A.Chhabria, Dr.R.V.Dharaskar, Dr.V.M.Thakare
 Asst.Prof.GHRCE,NAGPUR, Director,Nanded , HOD,Amravati University

Abstract—Different types of gesture recognition using Gaussian mixture model has been described in this paper. Human gesture has its specific meanings and is widely used for communications between deaf people. Hand gesture recognition using GMM, Speech recognition using GMM,SVM & fusion of gestures using GMM are described in this paper. Gaussian Mixture Model (GMM) helps us to determine the required region pixel clusters in the fused data. Advantages and comparison of GMM over other techniques like DTW,SVM are also discussed in this paper.

Index Terms— Multimodal system, Gesture recognition, Gaussian Mixture Model, Fusion Techniques

I. INTRODUCTION

A Gaussian Mixture Model (GMM) is a parametric probability density function represented as a weighted sum of Gaussian component densities. GMMs are commonly used as a parametric model of the probability distribution of continuous measurements or features such as vocal-tract related spectral features in a speaker recognition system. Training data is used for estimating GMM parameters using the iterative Expectation-Maximization (EM) algorithm or Maximum A Posteriori (MAP) estimation from a well-trained prior model.[1]

GMM [2] is a widely used statistical model in many applications of pattern recognition, which is often regarded as a versatile modeling tool as it can be used to approximate any probability density function (PDF) given a sufficient number of components, and impose only minimal assumptions about the modeled random variables. The advantage includes rigorous statistical basis, the possibility of encoding spatial, color, texture and motion features in a unified system, and the ability to trade off accuracy of representation against data volume.GMM thus estimates the mutative meaning of human gestures in a compact and precise manner. [2]

II. WORKING PRINCIPLE

One of the most widely used mixture modeling techniques is GMM [2] is. It is a simple model and is reasonably accurate when data are generated from a set of Gaussian distributions. Let $X_t = \{x_t, 1 \leq t \leq T_t\}$ denote the feature vectors for data points from the i -th class. They are modeled by a total number of J Gaussians as follows:

$$T_i \sim J$$

$$P(X^i | \theta_{GMM}) = \prod_{t=1}^T \sum_{j=1}^J P(z_j) P_{z_j}(x_t | \mu_j, \Sigma_j) \quad \text{Eq. (1)}$$

Where θ_{GMM} includes all the model parameters, i.e.

$$\{P(z_j), \mu_j, \Sigma_j, 1 \leq j \leq J\}. P_{z_j}(x_t | \mu_j, \Sigma_j) \quad \text{Eq. (2)}$$

is the Gaussian distribution for the j -th class, where μ_j is a mean vector and Σ_j a covariance matrix as:

$$P_{z_j}(x_t | \mu_j, \Sigma_j) = \frac{1}{(2\pi)^{D/2} |\Sigma_j|^{D/2}} \exp\left\{-\frac{1}{2}(x_t - \mu_j)^T \Sigma_j^{-1} (x_t - \mu_j)\right\} \quad \text{Eq. (3)}$$

where D is the dimension of the feature vector x_t . In order to reduce the size of parameter space, usually, Σ_j is set to be a diagonal matrix as

$$\text{diag}\{\sigma_{jd}^2: 1 \leq d \leq D\} \quad \text{Eq. (4)}$$

It can be seen from Equation (1) that the data points of a specific class are generated from multiple Gaussian models with an identical weight $P(z_j)$. We define

$$w_j = P(z_j) \quad \text{Eq. (5)}$$

In other words, an integrated Gaussian mixture model contains three basic parameters: Mixture weight, Mean vector and Covariance matrix, which can be represented as:

$$\lambda = \{w_j, \mu_j, \Sigma_j\} \quad \text{Eq.(6)}$$

where ω_j is the mixture weight, the mean vector is μ_j , and Σ_j the covariance matrix is Σ_j . We use λ to stand for every single image. Additionally, we use [2]

$$b_j(x) = P_{z_j}(x_t | \mu_j, \Sigma_j) \text{ and } \sum_{j=1}^J \omega_j = 1. \tag{Eq.7}$$

EM Algorithm

The expectation maximization (EM) algorithm is an iterative method for calculating maximum likelihood distribution parameter estimates from incomplete data (elements missing in feature vectors). The EM update equations are used which gives a procedure to iteratively maximize the log-likelihood of the training data given the model. The EM algorithm is a two step process:

Estimation Step in which current iteration values of the mixture are utilized to determine the values for the next iteration

$$\gamma(m, t) = \frac{w_m^{(i)} g(X_t, \mu_m^{(i)}, C_m^{(i)})}{\sum_{j=1}^M w_j^{(i)} g(X_t, \mu_j^{(i)}, C_j^{(i)})} \tag{Eq.8}$$

Predicted values are then maximized in Maximization step to obtain the real values for the next iteration.

$$\begin{aligned} \mu_m^{(i+1)} &= \frac{\sum_{t=1}^T \gamma_{m,t} X_t}{\sum_{t=1}^T \gamma_{m,t}} \\ W_m^{(i+1)} &= \sum_{t=1}^T \gamma_{m,t} \\ \lambda_{m,j}^{(i+1)} &= \frac{\sum_{t=1}^T \gamma_{m,t} (x_{t,j} - \mu_{m,j}^{(i+1)})^2}{\sum_{t=1}^T \gamma_{m,t}} \end{aligned} \tag{Eq.9}$$

EM algorithm is highly appreciated and well known for its numerical stabilities under a threshold values of λ min. Using the final w , μ and C re-estimated values, the value of LIGMM is calculated with respect to all the word models available with the recognition engine as:

$$L_{GMM} = \frac{1}{T} \sum_{t=1}^T \log P_{GM}(x_t) \tag{Eq.10}$$

III Hand Gesture Recognition using GMM:

It has been proved that human skin color seems to be an effective feature in many applications ranging from human face detection to hand detection and tracking, because it contains important information for detection. However, skin colors of different races in distinct context have different intensive clusters in normalized color space, especially for hand detection in various scenes. As been found, a single Gaussian distribution model is neither sufficient to model

hand skin color in hand detection, nor effective in general applications.

Thus a better way of dealing with the problem is Gaussian mixture model, GMM. For the purpose of improving hand skin detection, firstly images are handled using color correction, then train for a Gaussian mixture model, and finally detect images.

Frames from video sequence are analyzed and preprocessed. A Gaussian filter with a 5x5 mask must be applied to each frame to smooth the image. A mixture of K Gaussian distributions models each pixel to subtract the background and reach the hand segmentation. The Stauffer and Grimson [1] GMM model is used to subtract the background, in the RGB color space, and it is compared to different approaches from the same algorithm implemented by Power and Shoonees [3] and Kadew TraKuPong and Bowden [4].

GMM Algorithm Steps

a. Initializing K Gaussians per Pixel: [3]

Each new Gaussian K is created with the mean, variance and weight parameters equal to the current pixel value, the initial high variance and the low initial weight, respectively.

b. Checking the Standard Deviation Threshold:

Check if the pixel value (X_t) is within the 2.5 standard deviation of all existing K Gaussian distributions. Then calculate the standard deviation (σ) and the mean (μ) of each existing Gaussian to check the standard deviation criterion. There are different rules according to each GMM approach: Stauffer and Grimson [1] use the equation 11 and the equation 12 is employed in Power and Schoones [3], KadewTraKuPong and Bowden [4] methods:

$$|R - \mu_R| \leq 2.5\sigma \wedge |G - \mu_G| \leq 2.5\sigma \wedge |B - \mu_B| \leq 2.5\sigma \tag{Eq.11}$$

$$\left(\frac{(R - \mu_R)}{\sigma_R}\right)^2 + \left(\frac{(G - \mu_G)}{\sigma_G}\right)^2 + \left(\frac{(B - \mu_B)}{\sigma_B}\right)^2 \leq (2.5)^2$$

Eq. (12)

c. No Matching (Foreground pixel):

No distribution was found among the existing K Gaussian distributions.[3] The least probable distribution is replaced with a new distribution using the mean, variance and weight parameters equal to the current pixel value, the initial high variance and low initial weight, respectively. The least probable distribution is determined by the Gaussian distribution with the lowest ω/σ value.

d. Matching Found (Background pixel):

When a match is found among the existing K Gaussian distributions, the Gaussian parameters must be adjusted. The weights (ω) of all Gaussians are adjusted. The mean (μ) and the standard deviation (σ) are updated only for the matched Gaussian, while the unmatched Gaussians are not changed. The weights, means and deviations are updated where ρ is

calculated based on Stauffer and Grimson and using the equation 13 based on Power and Schoones and KadewTraKuPong and Bowden

$$\rho = \alpha / \omega_{i,j,k}$$

$$\rho = \alpha = \max(1/(N+1), 1/L)$$

Eq.[13]

Where N is the amount of matched distributions considered as a background, and L is the limit for the amount of matched distribution. Before the model reaches L matched distributions, the update equations consider $\alpha = 1/(N+1)$. After L distributions, the update equations consider $\alpha = 1/L$.

e. Choosing the Background Distribution:

After updating the parameters using the steps above, sorting of the Gaussians is done using ω/σ in descending order. Choose the first B distributions as a background model, that is, the sum of their weights (ω) is greater than T , as indicated in the equation 3.4: If the matched distribution is one of the first B distributions, the pixel is classified as a background pixel. Whether no distribution is found among the existing K Gaussian distributions or the pixels do not match any of the first B distributions, the pixel is classified as a foreground pixel.[3]

IV Speech Recognition using GMM & SVM /HMM

Joint Factor Analysis [5]

Joint factor analysis is a model used to treat the problem of speaker and session variability in GMM's. In this model, each speaker is represented by the means, covariance, and weights of a mixture of C multivariate diagonal-covariance Gaussian densities defined in some continuous feature space of dimension F . The GMM for a target speaker is obtained by adapting the Universal Background Model parameters (UBM). The UBM is trained using a large amount of data. In Joint Factor Analysis, the basic assumption is that a speaker and channel dependent supervector M can be decomposed into a sum of two supervectors: a speaker supervector s and a channel supervector c

$$M = s + c \quad \text{Eq.(14)}$$

where s and c are normally distributed. In [5], Kenny *et al.* described how the speaker dependent supervector and channel dependent supervector can be represented in low dimensional spaces. The first term in the right hand side of (14) is modeled by assuming that if s is the speaker supervector for a randomly chosen speaker then

$$s = m + vy + dz \quad \text{Eq. (15)}$$

where m is the speaker and channel independent supervector (UBM), d is diagonal matrix, v is a rectangular matrix of low rank and y and z are independent random vectors having standard

normal distributions. In other words, s is assumed to be normally distributed with mean m and covariance matrix $vv^* + d2$. The components of y are the speaker factors. The channel-dependent supervector c which represents the

channel effect in an utterance is assumed to be distributed according to $c = ux$ Eq.(16)

where u is a rectangular matrix of low rank, x is distributed with standard normal distribution. This is equivalent to saying that c is normally distributed with zero mean and covariance uu^* . The components of x are the channel factors in factor analysis modeling.

GMM-SVM's

This approach consists in the application of support vectors machines with GMM supervectors as input features for the speaker verification task. We refer to the supervectors as input features because, in the case of a general kernel $K(s, s')$ (whose arguments are pairs of supervectors), it is necessary to distinguish between input features and expanded features defined by the kernel mapping function

$$s \rightarrow K(s, \cdot) \quad \text{Eq.(17)}$$

Linear Kernel

The linear kernel that we used on GMM supervector space is derived from the distance between two GMMs based on Kullback-Leibler (KL) divergence. In the case of MAP adaptation with diagonal covariance matrices and when only the means of GMMs were adapted from the UBM, the weighted Euclidean distance between scaled version of GMM supervectors s and s' was given as follow:

$$D_e^2(s, s') = \sum_{i=1}^C w_i (s_i - s'_i) \Sigma_i^{-1} (s_i - s'_i)^t \quad \text{Eq.(18)}$$

where w_i and Σ_i are the i th UBM mixture weights and diagonal covariance matrix, s_i correspond to the mean of the Gaussian i of the speaker GMM. The linear kernel is defined as the corresponding inner product:

$$K_{lin}(s, s') = \sum_{i=1}^C \left(\sqrt{w_i} \Sigma_i^{-\frac{1}{2}} s_i \right) \left(\sqrt{w_i} \Sigma_i^{-\frac{1}{2}} s'_i \right)^t \quad \text{Eq.(19)}$$

This kernel was proposed by Campbell et

Non Linear Kernel

The non linear kernel that we used is the exponential version the distance between two GMMs $D_e^2(s, s')$ given in (5). It is given by the following equation.

$$K_{nonlin}(s, s') = e^{-D_e^2(s, s')} \quad \text{Eq.(20)}$$

This kernel was proposed first by Dehak and Chollet in [5]. The non linear kernel is equivalent to the Gaussian kernel defined on the GMMs supervector space. The corresponding expanded feature space is infinite-dimensional. The feature mapping function $\hat{A}(\cdot)$ is:

$$s \mapsto \phi(s) = K(s, \cdot) = e^{-\frac{\|s - \cdot\|^2}{2\sigma^2}} \quad \text{Eq.(21)}$$

The GMM/HMM Hybrid Model

The GMM/HMM hybrid model has the ability to find the joint maximum probability among all possible reference words W given the observation sequence O. In real case, the combination of the GMMs and the HMMs with a weighted coefficient may be a good scheme because of the difference in training methods. The *i*th speaker independent GMM produces likelihood LiGMM, I = 1, 2,..., W, where W is the number of words. The *i*th speaker independent HMM also produces likelihood LiHMM, I = 1, 2,..., W. All these likelihood values are passed to the so called likelihood decision block, where they are transformed into the new combined likelihood L'(W):

$$L'(W) = (1 - x(W))L_{GMM}^i + x(W)L_{HMM}^i \quad \text{Eq.(22)}$$

where x(W) denotes a weighting coefficient.

The value of x is calculated during training of the Hybrid Model. In Hybrid Testing, the subset of training data is used and it's HMM & GMM likelihood values are calculated which are combined using weighing coefficient. Static values of weighted coefficient are also used in order to get higher recognition rate.

V Fusion using GMM

The four levels of fusion are: (1) Sensor level,(2) Feature level, (3) Match score level, and (4) Decision level. For a Multimodal system, sensor level fusion is not advisable because it requires a compatible data for fusion, which is seldom the case with sensors. Fusion at the feature level is also not always possible because the feature sets used by different modalities may either be inaccessible or incompatible. At the decision level, very less amount of information is available for fusion, hence not widely used. Score level fusion offers the best trade-off in terms of the information content and the ease in accessing and combining matching score.[4]

Score fusion techniques can be divided into four categories: combination approach fusion, transformation based score fusion, density based score fusion and classifier based score fusion out of which Density based score fusion is chosen since it provides comparatively less error rates.[4]

Density-based score fusion approach is based on the likelihood ratio test and it requires explicit estimation of genuine and impostor match score densities. Density estimation can be done either by parametric or non-parametric methods. In parametric density estimation techniques, the form of the density function is assumed to be known and only the parameters of this density function are estimated from the training data. On the other hand, non-parametric techniques do not assume any standard form for the density function and are essentially data driven. This approach requires an

explicit estimation of genuine and impostor match score densities. The match score densities are obtained using GMM. The parameters required for GMM are obtained using Expectation Maximization (EM).To decide a score vector as genuine or impostor, the optimal method is Likelihood ratio test, according to Neyman-Pearson theorem.

After estimating the densities, the probabilities are computed and then decision rules are applied to make a decision. With accurate score densities, Density based approach has the advantage of achieving optimal performance at any desired False Acceptance Rate (FAR)

The extracted information from each modality fingerprint, iris and face are fused at the match score level using a density based score level fusion, GMM followed by the Likelihood ratio test. GMM parameters are estimated from training data using the iterative Expectation-Maximization (EM) algorithm.[8]

Gaussian Density is not appropriate for estimating match score density because the distribution usually has a long tail and more than one mode. So we go for Mixture of Gaussian Densities. A mixture model is a probabilistic model which assumes the underlying data to belong to a mixture distribution. In a mixture distribution, its density function is just a convex combination (a linear combination in which all coefficients or weights sum to one) of other probability density functions.

$$p(x) = w_1 p_1(x) + w_2 p_2(x) + \dots + w_n p_n(x) \quad \text{Eq.(23)}$$

The individual pi(x) density functions that are combined to make the mixture density p(x) are called the mixture components, and the weights w1, w2,...,wn associated with each component are called the mixture weights or mixture coefficients.

$$p_i = \phi^N(x | \mu_i, \Sigma_i)$$

$$\phi^N(x | \mu, \Sigma) = \frac{1}{(2\pi)^{N/2} |\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)} \quad \text{Eq.(24)}$$

ϕ^N denotes the N-variate Gaussian Density with mean vector μ and covariance matrix Σ .

Based on the this we obtain the genuine (*fgen(x)*) and impostor (*fimp(x)*) score densities,

$$f_{gen}(x) = \sum_{i=1}^{n_{gen}} (w_{gen,i} \phi^N(x | \mu_{gen,i}, \Sigma_{gen,i}))$$

$$f_{imp}(x) = \sum_{i=1}^{n_{imp}} (w_{imp,i} \phi^N(x | \mu_{imp,i}, \Sigma_{imp,i})) \quad \text{Eq.(25)}$$

where *n_{gen}* and *n_{imp}* are the number of mixture components used to model the genuine and impostor score densities respectively, *w_{gen,i}* and *w_{imp,i}* is the weight assigned to the *i*th

mixture component in $f_{gen}(x)$ and $f_{imp}(x)$ respectively.

$$\sum_{l=1}^{N_{gen}} W_{gen,l} = \sum_{l=1}^{N_{imp}} W_{imp,l} = 1$$

Eq.(26)

In order to find the component parameters μ and Σ , we use Expectation Maximization (EM) Algorithm.

EXPECTATION MAXIMIZATION IN FUSION

This algorithm is also robust to initialization of parameter values (mean vectors and covariance matrices) and to outliers.[4]

This is achieved by adding a small value (regularization factor) to the diagonal of the covariance matrices. The actual value of this variance does not affect the performance as long as it is insignificant compared to the variance of the continuous components in the match score distribution. The algorithm is

- Making an initial guess of the parameter vector:

This involves randomly selecting k objects to represent the cluster means or centres (as in k -means partitioning), as well as making guesses for the additional parameters.

- Iteratively refine the parameters (or clusters) based on the following two steps:

(a) Expectation Step (E-step): Assign each object x_i to cluster C_k with the probability

$$P(x_i \in C_k) = p(C_k | x_i) = \frac{p(C_k)p(x_i | C_k)}{p(x_i)}$$

Eq.(27)

Where $p(x_i | C_k) = \mathcal{O}(x_i | \mu_i, \Sigma_i)$ follows Gaussian distribution. In other words, this step calculates the probability of cluster membership of object x_i , for each of the clusters. These probabilities are the expected cluster memberships for object x_i .

(b) Maximization Step (M-step): Use the probability estimates from the above step to refine the model parameters so that it maximizes the expected likelihood. Repeat the E and M step until the parameters do not change or if the change in the parameters is below a specified threshold.[4]

LIKELIHOOD RATIO TEST

After obtaining the genuine and imposter score densities, Neyman-Pearson theorem is used to make accept or reject decision.[4]

$$LR(x) = \frac{f_{gen}(x)}{f_{imp}(x)}$$

Eq.(28)

X is assigned to genuine class if $LR(x) > f_{gen}(x)$, where $f_{gen}(x)$ is the threshold decided based upon the specified FAR.[4]

VI Comparison of GMM with other Techniques:

Single Gaussian model, & the GMM can capture more complex variations caused by the difference of human races, gender, and etc.

TGMR: *Time-dependent Gaussian Mixture Regression* [6] is based on our previous work, where time is used as an explicit input variable. The demonstrations are first aligned in time through *Dynamic Time Warping* (DTW), see [6] for details. Then, the distribution of temporal and spatial variables $\{t, x, x'\}$ is encoded in a *Gaussian Mixture Model* (GMM). At each time step during the reproduction process, a desired position \hat{x} and a desired velocity \hat{x}' are then retrieved through GMR by estimating $P(x, x' | t)$. The controller used by the robot to reproduce the skill is the mass-spring damper system .

LWR: *Locally Weighted Regression* [6] is a memory based probabilistic approach. It is used here to estimate at each time step a desired position \hat{x} and a desired velocity \hat{x}' . Each data point of the dataset participates in the estimation of the solution by using a Gaussian kernel with fixed diagonal covariance matrix centered at the current position to weight the influence of each data point.

LWPR: *Locally Weighted Projection Regression* is an incremental regression algorithm that performs piecewise linear function approximation [6]. The algorithm does not require storage of the training data and has been proved to be efficient in a variety of robot learning tasks including high dimensional data. We use here an implementation of LWPR with the input space defined by a set of receptive fields with full covariance matrices. By detecting locally redundant or irrelevant input dimensions, the method locally reduces the dimensionality of the input data by finding local projections through *Partial Least Squares* (PLS) regression. The learning parameters have been set based on the recommendations provided in [6]. During reproduction, LWPR is used at each iteration to estimate a desired velocity \hat{x}' , given the current position x . The receptive fields are then used to determine a desired position \hat{x} in a similar manner to the methods above.

DMP : The *Dynamic Movement Primitives* approach was originally proposed by Ijspeert *et al* [6]. The method allows a target to be reached by modulating a set of mass-spring damper systems. This allows a particular path to be allowed with the guarantee that the velocity vanishes at the end of the movement. A phase variable acts as a decay term to ensure that the system asymptotically converges to a reaching point.[6]

SVM : [7] The GMM classifier is simple to design and to compute its parameters. Recently a new machine learning classifier has been applied on different problems. This classifier is called the Support Vector Machine (SVM). The SVM works counter intuitive to how previous systems have been designed. The purpose of using feature-sets is motivated

by the desire to reduce the dimensionality of the data so that it can be manageable. The SVM increases the dimensionality of the data so that it can separate it better in some higher dimensional space where a simple hyper plane can be used to classify the data. This contrast with the GMM which is essentially based on vector quantization techniques where the aim is to reduce the dimensionality of the data and find some few parameters that can be used to model a talker.

The SVM approach is generally more expensive to compute than the GMM models. SVM does outperform the GMM when the amount of training data is limited. However, with more data the GMM approach is better. This can be easily explained, since with GMM when more data is available, the estimates of model parameters are better, whereas in the SVM case more data may not change the location of support vectors. Accuracy of GMM is 83%. Accuracy of SVM is 60%

Dynamic Time Warping

Dynamic time warping is an algorithm for measuring similarity between two sequences which may vary in time or speed. In our experiment, we use the DTW techniques which propose by Sadaoki Furui. According to Furui theory, the training data are used as a initial template, and the testing data is time aligned by DTW. DTW is a method that allows a computer to find an optimal match between two given sequences. The average of the two patterns is then taken to produce a new template to which a third utterance is time aligned. This process is repeated until all the training utterances have been combined into a single template. Increasing amounts of training data, the DTW distance measure become hard to calculated due to the progressively information of speaker.

DTW suitable due with small fixed vocabulary system. Accuracy is 80%.[7]

Hidden Markov models:

Gaussian Mixture models— A probabilistic model of .Feature vectors associated with a speech sound.

Principled distance between test frame and set of template frames.

Hidden Markov models— A probabilistic model of Time evolution of feature vectors for a speech sound. Principled generalization of DTW.

Conclusion :

GMM is better as compared to SVM,HMM,DTW. as it provides improved performance over other existing methods, yet requiring only modest computational cost to complete the gesture recognition.

REFERENCES

- [1] Douglas Reynolds,"Gaussian Mixture Models",MIT Lincoln Laboratory, 244 Wood St., Lexington, MA 02140, USA
- [2] Jia Jia, Jianmin Jiang, Dong Wang EIMC, School of Informatics,University of Bradford, West Yorkshire BD7 1DP, UK Paper titled,Recognition of Hand Gesture Based On Gaussian Mixture Model
- [3] Hebert Luchetti Ribeiro Adilson Gonzaga,"Hand Image

Segmentation in Video Sequence by GMM: a comparative Analysis", School of Engineering at São Carlos University of São Paulo -USP University of São Paulo – USP São Carlos, São Paulo, Brazil

- [4] Feature Extraction for Multimodal Biometric and Study of Fusion Using Gaussian Mixture Model , Proceedings of the International Conference on Pattern Recognition, Informatics and Medical Engineering , March 21-23, 2012
- [5] Comparison Between Factor Analysis and GMM Support Vector Machines for Speaker Verification, Najim Dehak1 R'eda Dehak2, Patrick Kenny1, Pierre Dumouchel
- [6] Paper titled, "Evaluation of a probabilistic approach to learn and reproduce gestures by imitation " published at 2010 IEEE International Conference on Robotics and Automation Anchorage Convention District May 3-8, 2010, Anchorage, Alaska, USA
- [7] Loh Mun Yee,Abdul Manan Ahmad,Faculty of Computer Science & Information System University of Technology Malaysia Skudai, 81310 Johor Darul Ta'zim, Malaysia "Comparative Study of Speaker Recognition Methods: DTW, GMM and SVM".