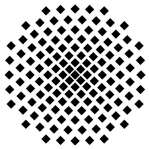

University of Stuttgart
Institute for Signal Processing and System Theory
Professor Dr.-Ing. B. Yang

# PÜL on Statistical Signal Processing

## Prostate Cancer Segmentation and Speaker Identification

Author: Nithin Kumara, Narayanaswamy Teekaramanaa (3212235)
Srirama Dharma Teja, Vattem (3213807)
Bineeta, Saikia (3213823)

Date of work begin: 23.10.2017
Date of submission: 09.02.2018

Supervisor: Lukas Mauch

# Contents

# 1 Introduction

In this lab we are asked to implement the already learned methodologies in "Detection and Pattern Recognition" lecture for classification. We will be implementing classification algorithms using supervised learning methodologies in two tasks namely

1. Prostate Cancer Segmentation
2. Speaker Recognition

## 1.1 Prostate Cancer Segmentation

In this task the challenge is to detect the cancerous region from the images of prostate taken through various methodologies. We are given different types of 3D images taken by five different methodologies namely:

1. T2 weighted MR images
2. ADC (Apparent diffusion Coefficient) map
3. K-Trans map
4. K-ep map
5. PET map

We have to use these five different types of 3-D images and detect the cancerous portion in the prostate region of the patients. A total of 14 patients are given among which the first 11 patients are to be used for training and the rest for validation. The voxel values are also labelled by two experts based on which classification has to be done.

## 1.2 Speaker Recognition

In this task we have to perform speaker recognition. We are given a Universal Background Model (UBM) which is trained using training data of TIMIT database. The task here is to adapt the UBM to the test speakers and our own voice samples. For each test speaker we get one adapted model. Later during prediction we predict new audio sample with these adapted models and estimate the speaker.

# 2 Task 1: Prostate Cancer Segmentation

In this section we explain our pipeline which has been used to perform the prostate cancer segmentation. With the already provided pipeline a few steps have been added in order to perform outlier detection and dimensionality reduction. Figure 2.1 gives us an idea of the entire pipeline.

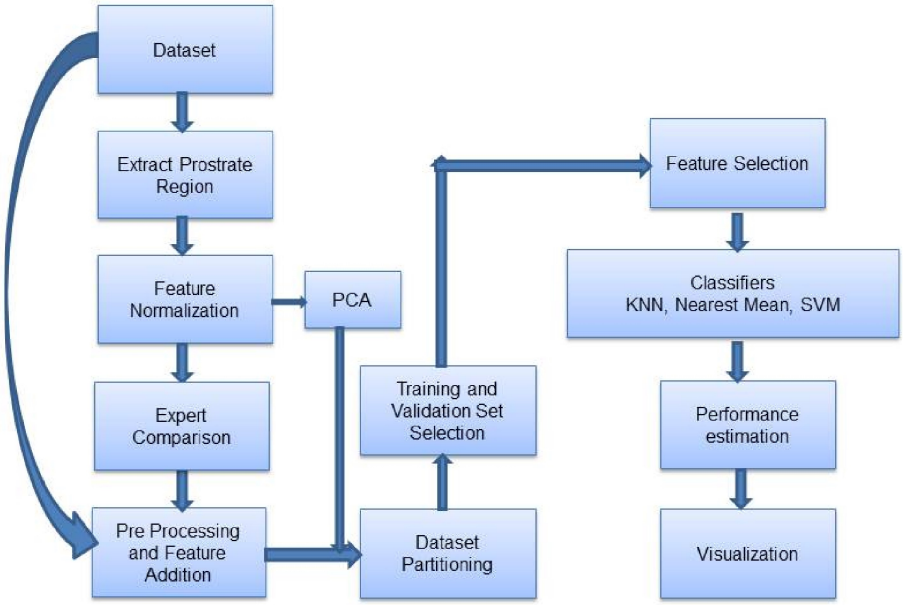


Figure 2.1: The Pipeline

## 2.1 Dataset/Prostate Region Extraction

The provided dataset consists of the non-prostate region as well. Hence, the prostate region of all the patients has been extracted first to form one database consisting of voxel information of the entire prostate region with location information, expert labels and patient id.

## 2.2 Feature Normalization

Three different techniques are used to normalize the dataset. The methods used are as below [10]:

1. Mean and Variance: Here the dimension of each feature vector has been scaled to zero mean and unit variance as below:

$$\tilde{x_k} = \frac{x_k - \mu_k}{\sigma_k} \tag{2.1}$$

2. Min – Max Normalization: In this technique each feature vector has been normalized to unit dynamic range of [0,1]

$$\tilde{x_k} = \frac{x_k - min(x_k)}{max(x_k) - min(x_k)} \tag{2.2}$$

3. Unit Vector Normalization: Here each feature vector has been normalized to unit vector length

$$\underline{\tilde{x_k}} = \frac{\underline{x}}{|\underline{x}|} \tag{2.3}$$

Considering the above three methods, method 1 gave us the best results with SVM and kNN classifiers whereas for Nearest Mean, method 2 gave the best results.

## 2.3 Expert Comparison

As we saw that both the experts contradict in their labelling for some of the voxel points, hence we decided to find the most accurate expert by comparing them with the histological labels. For Patient 12 to 14 we compared the histological labels with that of the expert labels and found the accuracy as below:

1. Expert A =84.905
2. Expert B =83.467

As Expert A has higher accuracy, we considered the labels of Expert A for training and validation.

## 2.4 Pre-Processing and Feature Addition

In order to understand the position of the voxels in the prostate we should have the neighbourhood information around the voxels. Hence, considering the cubic structure of the 3-D images the label values of the neighbourhood voxels were added and had been taken as frequency value. That is, if we consider a voxel, the labels of its neighbours can be either 1 or 2. Hence, considering its position parameters we added its neighbourhood labels and found that the voxels surrounding the cancerous region can have a max value of 54 and the

voxels surrounded by all non-cancerous region can have a value of 27. Whereas, the voxels at the edge i.e. at the boundary or not surrounded by all non-cancerous region will have lower value than 27.This is a way of adding neighborhood information to each voxel and when included along with other features becomes a good feature in distinguishing cancerous and non-cancerous voxels.

Adding location information as rectangular coordinate is not a good feature. Hence we convert the rectangular coordinates to spherical coordinates where we get radius, azimuthal and elevation. We take azimuthal and elevation which are angles in spherical coordinates that point to the direction of the point from origin. This is a good feature that embeds location information.

## 2.5 PCA

In multivariate data analysis, PCA is a very common and powerful technique from linear algebra used to reduce high dimensional data to low dimension. It transforms a number of correlated variables in high dimension to new set of variables in low dimension called Principal Components and at the same time ensures the variance information of the data in low dimension to a great extent. The reasons for the reduction to lower dimensions could be to visualize the data in 2or 3 dimensions or to reduce the computational complexity.

In this task PCA would be carried out to visualize the nature of our dataset (linearly/non-linearly separable) which is not possible in the original 5-dimensional feature space. PCA is carried out on the covariance matrix of the data in 5-dimensional space.

Solution to the optimization problem of PCA gives us principal components which are eigen vectors of covariance matrix. Highest variance of data lies in the direction of eigen vector corresponding to highest eigen value and next highest variance in the direction of 2nd highest eigen vector and so on. In order to reduce the dimensions, we take eigen vectors corresponding to highest eigen values as basis vectors and transform our original feature space according to this newly formed basis. For the purpose of visualization, we consider 2 eigen vectors corresponding to highest 2 eigen values. Reducing the feature space to 2 dimensions and plotting the data points reveal information about the data. It was observed that the data is highly complex in nature with the cancerous and non-cancerous data clouds completely overlapping. Thus, it was very difficult to separate the data set into two parts.

In our case, PCA did not significantly improve computational complexity as the dimensions were reduced from 5 to 2. Although it was observed that the models were performing much better on test data when trained on PCA dataset. One reason that we understood could be the enrichment of data with PCA and only the most useful information is contained in these 2 dimensions. [8]

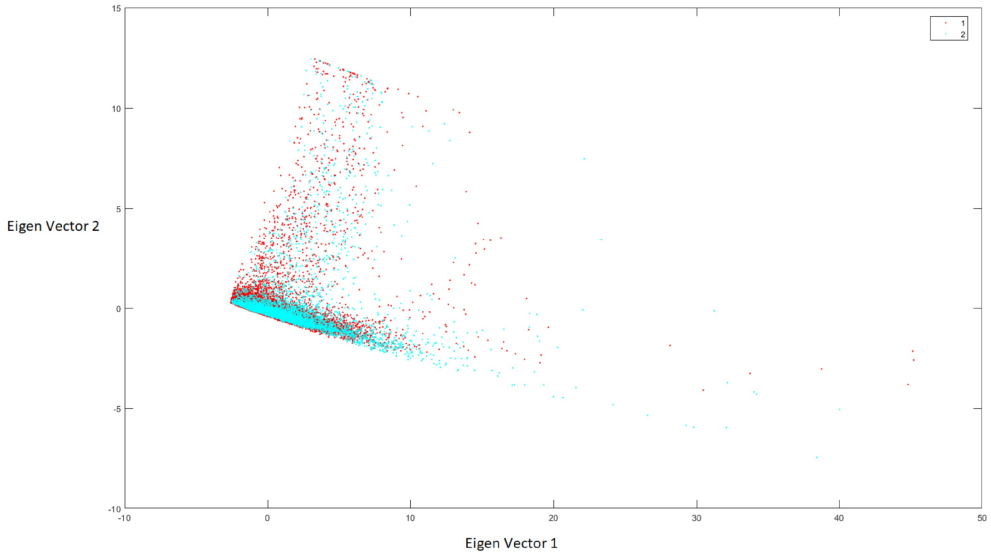


Figure 2.2: Features reduced to 2 using PCA

## 2.6 Training and Validation Set Selection

Another important task is to reduce the training set. Various methods have been used to reduce the training set in order to get the best of the training data for our model.

1. **Combination of K-Means clustering and neighbourhood based information**: In this method, first k means clustering algorithm was run and the dataset was divided into two parts. It was observed that the first cluster has majority of class 1 and second cluster has majority of class 2. The next step was to exploit the close nature of voxels present together. In first cluster, all the groups of 4 voxels occurring together with class label 1 were merged into one data point by taking the mean. Similarly in second cluster, all the groups of 4 voxels occurring together with class label 2 were merged into one data point by taking mean. Such groups of 4 voxels were discovered by traversing the clusters sequentially. This resulted in reduction of dataset by 50%. [11]

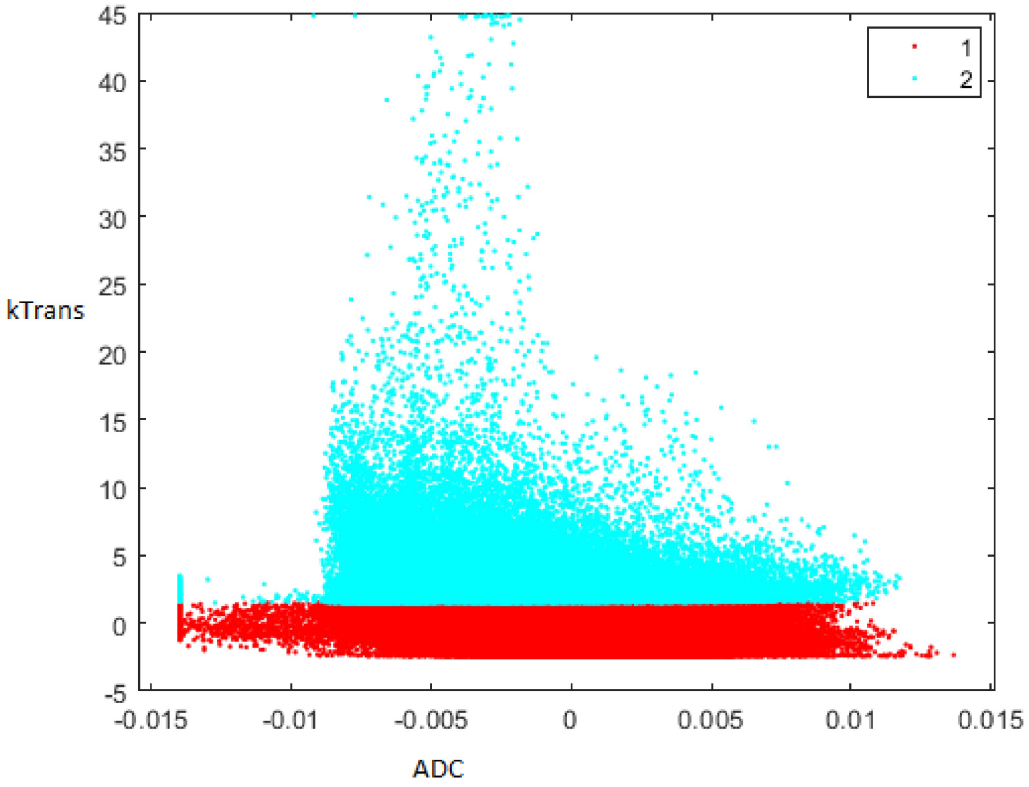


Figure 2.3: Visualization of the Cancerous and Non cancerous through k means clustering

2. **Random Selection 10% data**: We reduced the training dataset by randomly selecting every 10th data of the training dataset. This didn't alter any physical properties of the voxels hence gave us the best of the results.

3. **Random selection using "randdata" function**: We also reduced the "randdata" function to reduce the training dataset. This gave us almost the same results as by selecting every 10th data of the training dataset.

4. **Frequency selection**: We added the neighbourhood information considering frequency as the parameter as referred in section 2.4. We reduced the training dataset by plotting the frequencies of the training dataset first. It can be observed that around 250000 voxels have frequency 27. This represents that, 250000 voxels are surrounded by all non-cancerous prostate voxels. Whereas, very few voxels have a frequency ranging from 0-27. As explained in section 2.4, we saw that the voxels that are at the boundary or not surrounded by all non-cancerous region will have a lower value. Hence, in order to optimally reduce the training dataset we selected the voxels which are at the boundary and are surrounded by all cancerous voxels.

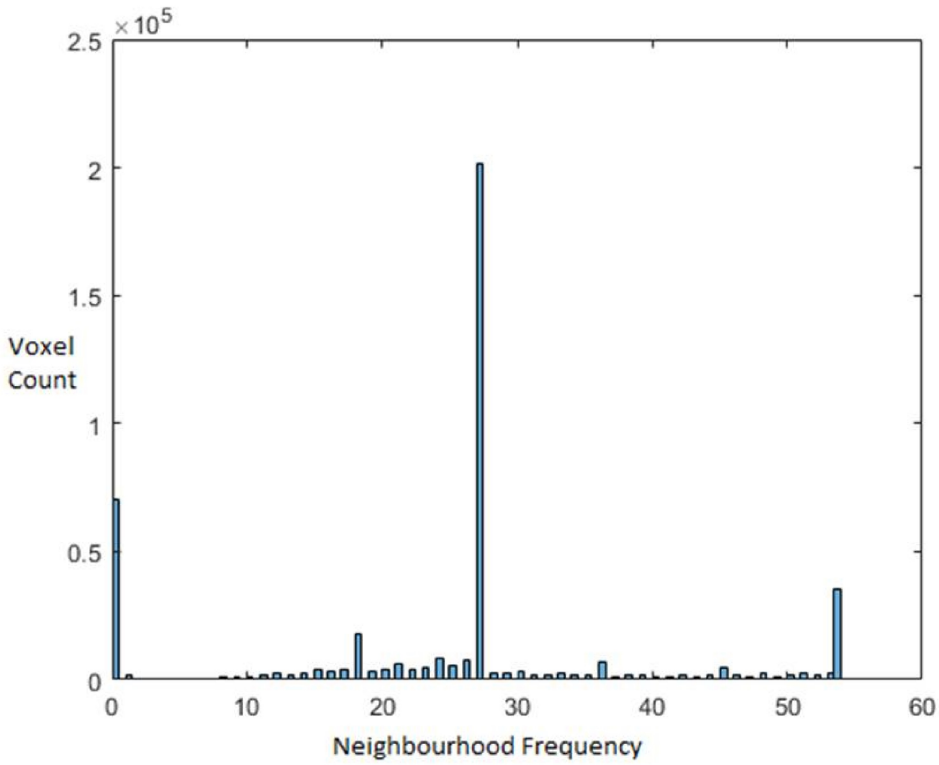


Figure 2.4: The Voxel counts with respect to the Neighbourhood frequency can be vividly seen

## 2.7 Feature Selection

Initially all the 5 features i.e. the 5 different types of images are taken and fed to our model. Later through iterations between two particular features i.e. two different types of images, we found that KTrans and ADC provides the best results. Hence KTrans and ADC are chosen as features to model the classifiers.

Moreover, we added frequency (as per section 2.2) as another feature to obtain our results. We first trained our model using the five features; thereafter we added the 6th feature the frequency as a feature and trained our model and there after validated the model that also had the frequency information.

## 2.8 Classifiers

The three different classifiers have been implemented on the normalized and reduced datasets to classify the test dataset into cancerous and non cancerous region. [9]

1. **Nearest Mean Classifier**

In the nearest mean classifier we measure the Mahalanobis distance of the validation data from the means of the two classes namely, cancerous class and the non-cancerous class. The cancerous and the non-cancerous classes are first separated using the training dataset and mean of both the classes are measured by:

$$\underline{\hat{\mu}_k} = \frac{1}{N} \sum_{n=1}^{N_k} \underline{x}_{k,n} \tag{2.4}$$

where k=2, $\underline{\hat{\mu}_k}$ represents the estimated class mean for each class and $\underline{x}_{k,n}$ are the feature vectors for each class from the training dataset. Thereafter the Mahalanobis distance has been calculated of the validation dataset for each feature vector $\underline{x}$ from the centres of both the classes :

$$\mathrm{D}(\underline{x}, \underline{\hat{\mu}_k}) = \sqrt{(\underline{x} - \underline{\hat{\mu}_k})^T \hat{C}_k^{-1} (\underline{x} - \underline{\hat{\mu}_k})} \tag{2.5}$$

The feature vectors having the minimum Mahalanobis distance from the two centres belong to that class. This means minimal $\mathrm{D}(\underline{x}, \underline{\hat{\mu}_k})$ gives the class of the validation feature vectors We use Mahalanobis distance instead of Euclidean distance as, our data set have non linear boundaries which cannot be realized by Euclidean distance.

2. **kNN Classifier**

   In the kNN classifer there is no computation during training. The classification of the feature vectors will directly depend on the k nearest neighbouring features of the training dataset. That is, amongst the $\underline{x}$ feature vector from the validation data set, we find the k nearest feature vectors from the training dataset. Among the k nearest neighbours, we decide for the class which is represented maximum number of times by the feature vectors. The k neighbours are found by measuring the Euclidean distances from the feature vectors of the validation dataset to the feature vectors of the test dataset.

   Hence the values of k will control the complexity of the decision boundary. For k being too small, the decision boundary will be complex and it tends to have over fitting. Whereas for k being large the boundary is smoother. In our case, we have iterated number of times to find the best k value to give better accuracy rates. We found that with k in between 85 to 100, gives us the best accuracy. We chose k=85 for our results as there was not much variation for the k values until 100.

3. **SVM**

   Support Vector Machines (SVM) is a binary classifier that can be used to separate non-linearly distributed data. It is a convex optimization problem with inequality constraints. The solution to this optimization problem gives us all the support vectors within the dataset. While computing the class of a new data point we only need these support vectors and all the data points are not required. Support vector machine is very robust to outliers as it does not depend on all the data points. It uses kernel trick in order to carry out feature transform to a higher dimensional space without actually knowing the transformation function. In order to compensate for the class labels on the wrong side of decision boundary we use a trade-off constant that penalizes the

points on the wrong side of decision boundary.
In this task we use the implementation of svm available in libsvm package. Radial basis function is used as kernel function. The hyperparameters that needs to be initialized are radial basis function coefficient g and trade off constant C. The hyperparameters have been selected naively by randomly choosing few values and selecting the one which gave best accuracy. In our case C = 1 and g = 0.1 were chosen. One efficient way to do hyperparameter optimization would be to use Bayesian hyperparameter optimization.

## 2.9 Approach

We implemented the entire pipeline in iterations to obtain our results. Below a map is given where we can see that for every normalization techniques (as in section 2.2), we did the training set reduction (2.6). Also for every training set reduction techniques, we implemented the Classifiers (2.8). Doing so, we could find the best results that our model can predict in order to classify the cancerous and the non-cancerous voxels. Below in fig 2.11 the entire flow of our execution process can be seen

| | | | |
|---|---|---|---|
| Dataset | Min-Max Normalisation | Combination of K-Means Clustering Neighbourhood based elimination | SVM |
| | | | KNN |
| | | | Nearest Mean |
| | | Random Selection 10% data | SVM |
| | | | KNN |
| | | | Nearest Mean |
| | | Random Selection randdata funct | SVM |
| | | | KNN |
| | | | Nearest Mean |
| | | Frequency Selection | SVM |
| | | | KNN |
| | | | Nearest Mean |
| | | PCA | SVM |
| | | | KNN |
| | | | Nearest Mean |
| | Magnitude Normalisation | Combination of K-Means Clustering Neighbourhood based elimination | SVM |
| | | | KNN |
| | | | Nearest Mean |
| | | Random Selection 10% data | SVM |
| | | | KNN |
| | | | Nearest Mean |
| | | Random Selection randdata funct | SVM |
| | | | KNN |
| | | | Nearest Mean |
| | | Frequency Selection | SVM |
| | | | KNN |
| | | | Nearest Mean |
| | | PCA | SVM |
| | | | KNN |
| | | | Nearest Mean |
| | Mean-Variance Normalisation | Combination of K-Means Clustering Neighbourhood based elimination | SVM |
| | | | KNN |
| | | | Nearest Mean |
| | | Random Selection 10% data | SVM |
| | | | KNN |
| | | | Nearest Mean |
| | | Random Selection randdata funct | SVM |
| | | | KNN |
| | | | Nearest Mean |
| | | Frequency Selection | SVM |
| | | | KNN |
| | | | Nearest Mean |
| | | PCA | SVM |
| | | | KNN |
| | | | Nearest Mean |

Figure 2.5: The execution flow

## 2.10 Results and Conclusion

1. **Visualization of prostate region of all training patients**

   In Fig 2.6 we can see the prostate region of the 11 patients we used for training

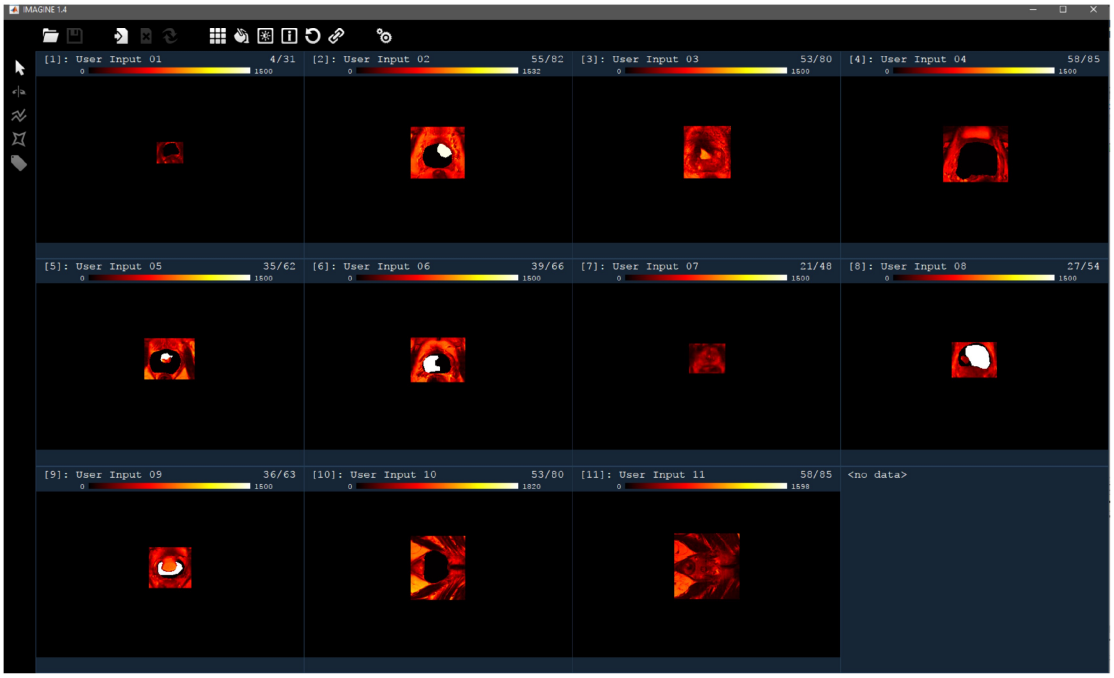


Figure 2.6: The Prostate Region of all Patients

2. **Prostate Data Visualization**

In Fig 2.7 we can see that the prostate data and the cancerous data corresponding to two features at a time are over lapping. It can be inferred that boundary of the two classes is not well defined

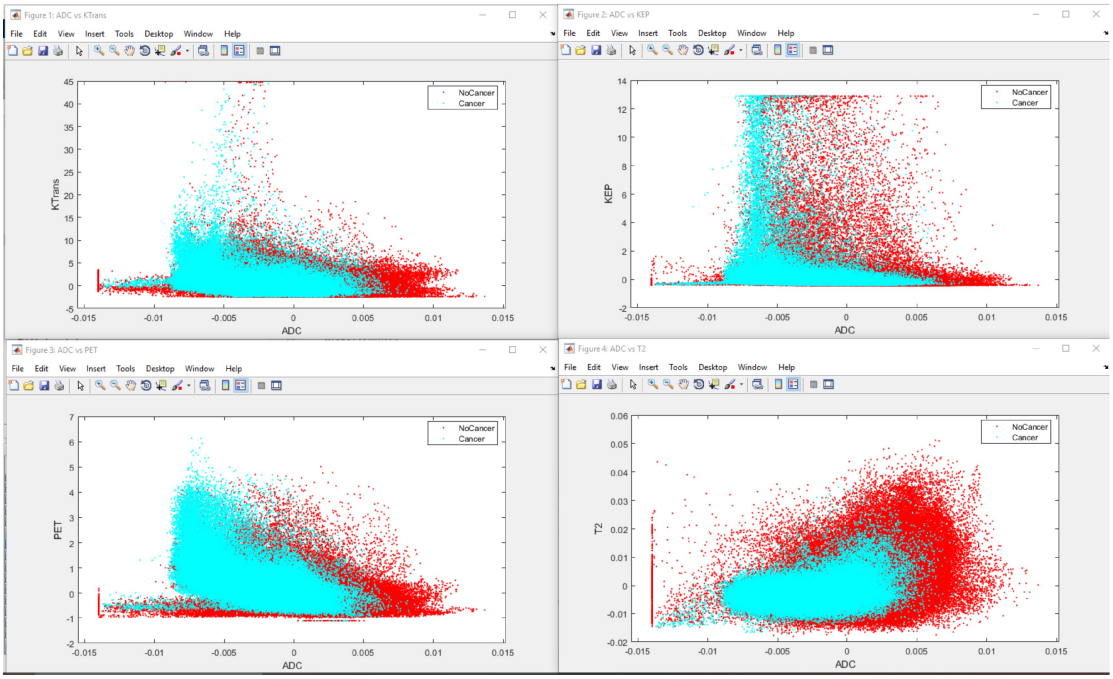


Figure 2.7: The Prostate data Visualization of cancerous and non cancerous data

3. **Comparison of reduced dataset**

In Fig 2.8 the detection result of the reduced training data set using 10% of the training data and a kNN has been compared to that of the validation dataset labelled by the experts

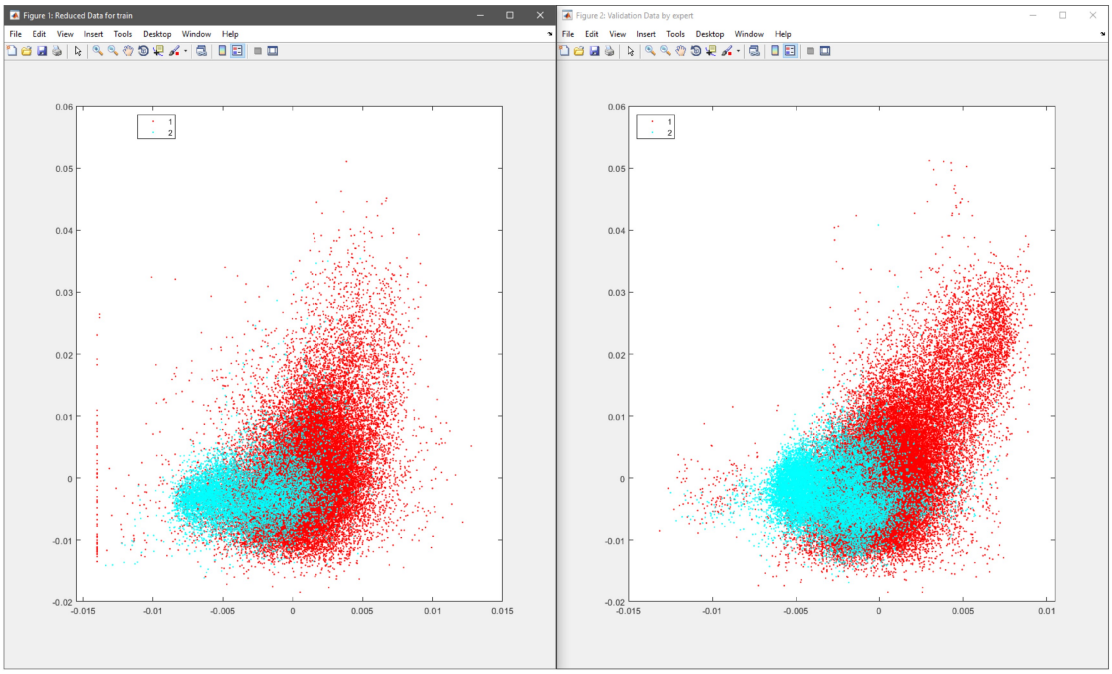


Figure 2.8: Comparison of reduced training dataset to that of validation dataset

4. **Detection results of kNN Classifier**

   In Fig 2.9 and 2.10 the visualization of the cancerous and the non cancerous part of the prostate region classified by kNN classifier can be seen for patient 12 and 14

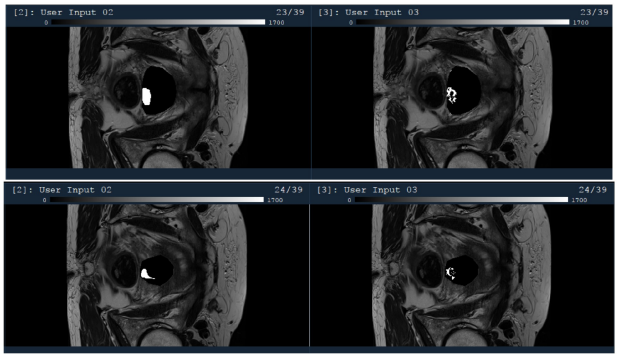


Figure 2.9: kNN results for Patient 12 with feature reduction using PCA and in comparison to Histological Labels

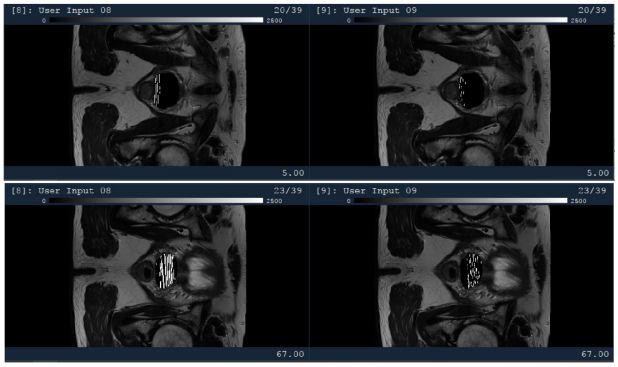


Figure 2.10: kNN results for Patient 14 with feature reduction using PCA and in comparison to Histological Labels

### 2.10.1 Accuracy

As mentioned in section 2.9 we performed all the combinations as mentioned in section 2.11 and came up with the best results as shown below:

| Classifier | Approach | Percentage |
|---|---|---|
| KNN | Random 10th with Mean-Variance Normalisation | 84.60% |
| | PCA | 80.69% |
| Nearest Mean | Random 10th with Min max normalisation | 83.53% |
| | PCA | 79.85% |
| SVM | Random 10th with Mean-Variance Normalisation | 76.70% |
| | PCA | 81.39% |

Figure 2.11: The accuracies of the different Classifiers can be seen with respect to the normalization methods used

### 2.10.2 Issues faced

1. Un separable data: The dataset provided isn't separable i.e. the cancerous and the non-cancerous regions overlap. Hence training and thereby validation becomes difficult through such a dataset.
2. Computational Complexity of CNN: One way of reducing the training dataset was by implementing CNN(Condensed Nearest Neighbour)Algorithm didn't terminate even after execution time of 40 hours
3. SVM K-Means Clustering: Also we couldn't implement SVM K-Means Clustering for dataset reduction as the memory requirement for this algorithm was beyond our Machine's capability.
4. Training the dataset twice using frequency(2.4) as the another feature didn't give us any improvised results which is normally expected to be better.

# 3 Speaker Identification pipeline

The Speaker recognition task is done by the following pipeline stepsas illustrated in the Figure 3.1.

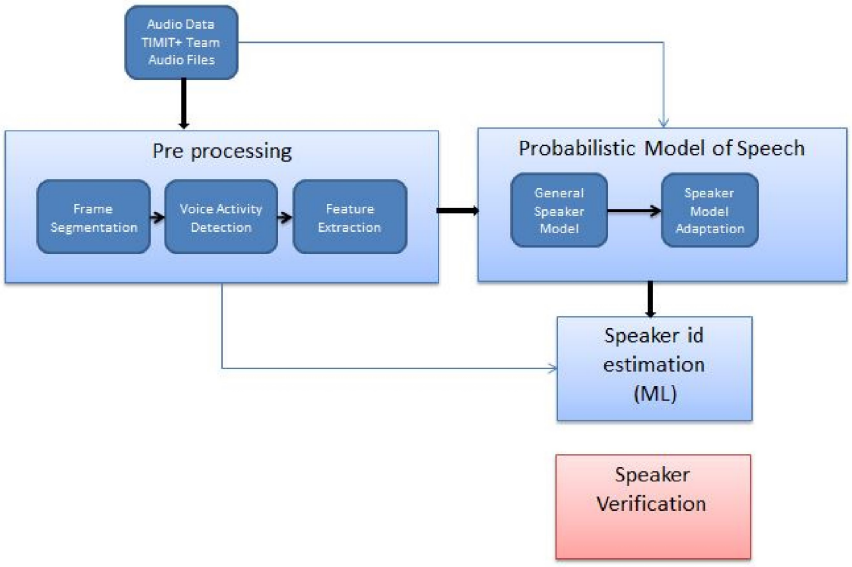


Figure 3.1: Speaker Identification block diagram

## 3.1 Frame Segmentation

For discriminating and identifying speech signals, small changes in the spectral characteristics are important clues. These changes can occur over very short time intervals. Hence, the audio data is divided into short overlapping frames with $t_{frame}$ = 20ms and $t_{feed}$ = 10ms, between two neighbouring frames. The signal in each frame is now assumed to be statistically stationary with the length of each frame, L = $f_s$ . $t_{frame}$ = 320 sample values of raw audio data x(n). Every $k^{th}$ frame's audio data is taken from x(n) as follows,

$$\vec{x_k} = \begin{bmatrix} x(k \cdot f_s \cdot t_{feed}) \\ x(k \cdot f_s \cdot t_{feed} + 1) \\ . \\ . \\ . \\ x(k \cdot f_s \cdot t_{feed} + L - 1) \end{bmatrix}, \qquad k = 0, 1, ..., K - 1 \tag{3.1}$$

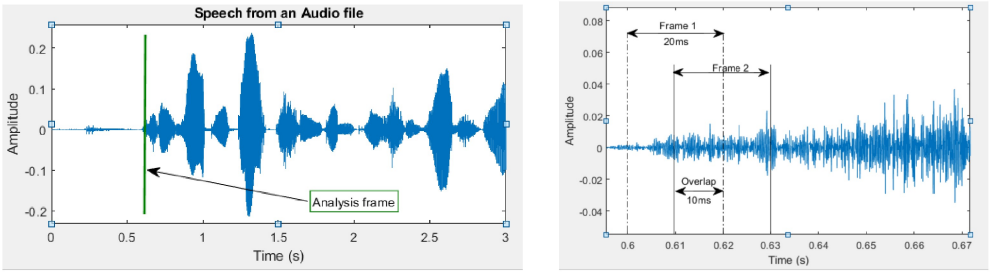


Figure 3.2: Image to the left is raw audio speech file and its corresponding analysis frame's segmentation is to the right.

where k is the number of frames:

$$K = \left[ \frac{OverallNumberOfSampleValues - L}{f_s.t_{\text{feed}}} \right] + 1 \tag{3.2}$$

Each audio file from the TIMIT database is read and stored as a matrix with dimensions (samples x Number of frames). If the last frame has less than 320 samples, then it is discarded in our implementation instead of padding zeros. Figure 3.2 illustrates the segmentation process.

## 3.2 Voice activity detection

All the frames extracted from section 3.2 may not contain the required voice activity. In this step we discard the frames which contains more pauses or only background noise. Every frame is now classified into voiced and unvoiced frames as follows,

- **Step 1 :** The speech signal power for the k-th frame is calculated by,

$$P(k) = \frac{1}{L} \sum_{i=0}^{L-1} x_k(i)^2 \tag{3.3}$$

- **Step 2 :** It is assumed that there is no voice activity during the first $t_n$= 100ms of each audio file. Hence, the Noise power $P_N$ is the average signal power in the first $K_{noise}$ frames.

$$K_{noise} = \left[ \frac{t_n}{t_{feed}} - 1 \right] = 9 \tag{3.4}$$

$$P_N = \frac{1}{K} \cdot \sum_{k=0}^{K_{noise}-1} P(k) \tag{3.5}$$

- **Step 3 :** Classification of all frames is as per the equation 3.6 .The value of $\gamma = 500$ is decided based on trial and error, making sure that sufficient frames are available for model training and the final test accuracy is good.

$$P_k \underset{H_0}{\overset{H_1}{\gtrless}} \gamma \cdot P_N \tag{3.6}$$

Figure 3.3 illustrates the output for an audio sample after implementing the above three steps.

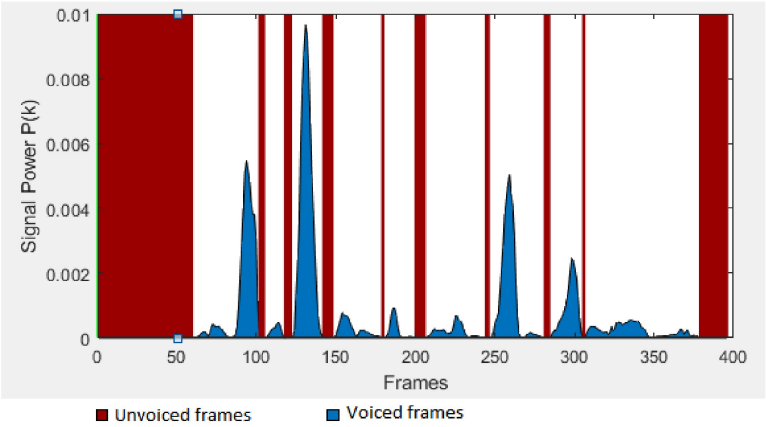


Figure 3.3: Voice activity detection

## 3.3 Feature extraction

The feature extraction is associated with the periphery in the brain's perception of speech. It has to be independent of the text spoken, background noise and also voice alterations due to illness, tiredness, mood that may affect the recognition performance REFMISSING. A speaker's voice is formed as per the excitation of larynx and alteration in the oral and nasal cavity. These characteristic properties are mapped in the voice spectrum to differentiate every speaker. Hence, each speaker will have a different periodogram based on the dominant frequencies used by the respective speaker.
For each frame we extract the Mel Frequency cepstral coefficients(MFCC) [2] as features which possess the information of formants i.e., each of several prominent bands of frequency that determine the phonetic quality of a vowel.

### 3.3.1 Windowing

Windowing is a method to amplitude modulate the input signal so that the spectral leakage is evened out . It reduces the amplitude of the samples at the beginning and end of the window, altering leakage. It uses a tapering function to smooth out the transitions. Here, we pass every $k^{th}$ frame through a Hamming window ($h_m(n)$) as per the equation 3.7.

$$x_{windowed}(k) = x_{voiced}(k) \cdot h_m(f_s \cdot t_{frame}), \quad 1 \leq k \leq L \tag{3.7}$$

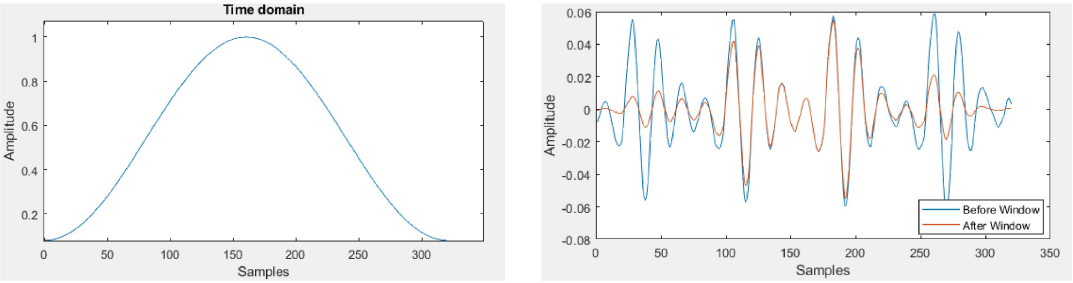


Figure 3.4: Left image indicates the hamming window and the right image illustrates the output after windowing.

### 3.3.2 Discrete Fourier Transform

Every frame is now analyzed in frequency domain by applying Discrete Fourier Transform (DFT),

$$S(k) = \sum_{n=1}^{N} x_{windowed}(k) \cdot e^{-j2\pi kn/N} \quad 1 \leq k \leq K \tag{3.8}$$

where, N is Number of samples per frame and K is the Length of Discrete Fourier Transform (DFT). The Figure below illustrates an output of the DFT,

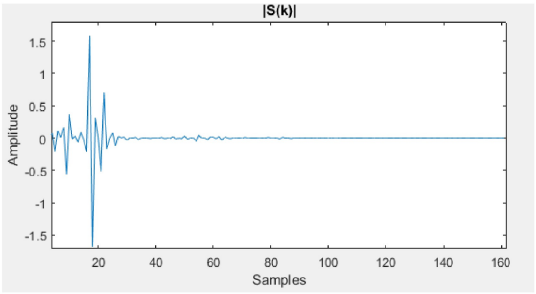


Figure 3.5: DFT of a frame

### 3.3.3 Mel Scale

The spectrum of each windowed voice frame is converted into Mel Spectrum as human ear senses the frequencies on a logarithmic scale defined by the Mel scale piecewise function as per equations3.9 and 3.10 [4]. The voice frequency bands typically ranges between 300 Hz to 3400 Hz. In this task we consider lower cutoff frequency as 0 Hz and upper cutoff frequency is $f_s/2 = 8kHz$ based on Nyquist–Shannon sampling theorem [3].

$$f_{mel} = \begin{cases} f & , f \leq 1kHz \\ 2595 \cdot log_{10}(1 + \frac{f}{700}) & , f > 1kHz \end{cases} \tag{3.9}$$

$$f_{mel}^{-1} = \begin{cases} f_{mel} & , f \leq 1kHz \\ 700 \cdot (10^{\frac{f_{mel}}{2595}} - 1) & , f > 1kHz \end{cases} \tag{3.10}$$

### 3.3.4 Mel Filter Bank

The purpose of a filter bank is to reduce the feature vector size significantly by considering only critical spectral parameters. Human ears cannot differentiate a small change in frequency at higher frequencies due to its logarithmic perception of sound. We first estimate the amount of energy that exists by considering periodogram bins at dominant frequencies in a frame. This is performed by a Mel filterbank as it applies Mel-frequency scaling to get better resolution at low frequencies and less at high frequency, mimicking a human ear [5].

We consider the cut off frequencies as per section 3.3.3 and create 22 triangular filter banks in mel scale. For this, we consider 24 linearly spaced frequency points $f_{melprojected}$ on the mel scale and project it on normal frequency from equations 3.9 and 3.10. The DFT bin numbers for the frequencies are,

$$f(i) = floor\left[(nfft + 1) \cdot \frac{f_{melprojected}(i)}{f_s}\right] \quad 1 \leq i \leq 24 \tag{3.11}$$

where, nfft is chosen to be 320 to obtain 160 as the highest DFT bin number.

The output of DFT is in complex plane. Hence, DFT of a frame with 320 samples is mirrored at mid point i.e., 160th sample. Therefore, we only consider 160 samples from the DFT output.
The triangular mel filter function is given by,

$$H_m(k) = \begin{cases} 0 & , k < f(m-1) \\ \dfrac{k - f(m-1)}{f(m) - f(m-1)} & , f(m-1) \leq k \leq f(m) \\ \dfrac{f(m+1) - k}{f(m+1) - f(m)} & , f(m) \leq k \leq f(m+1) \\ 0 & , k > f(m-1) \end{cases} \tag{3.12}$$
$$where, 1 \leq m \leq 22\ ;\ 1 \leq k \leq 160$$

The filters untill 1000 Hz are very narrow and they get wider as the frequencies get higher as we become less concerned about variations at higher frequencies, this is shown in Figure 3.6 .

We are only interested in roughly how much energy occurs at each spot. The magnitude coefficients of DFT are binned by correlating them with each triangular filter [6]. Here binning means that each DFT magnitude coefficient is multiplied by the corresponding filter gain and the results Y(k) are accumulated. Thus, each bin holds a weighted sum representing the spectral magnitude in that filterbank channel as shown in Figure 3.7.

$$Y(k) = |S(k)| \cdot H_m(k) \tag{3.13}$$

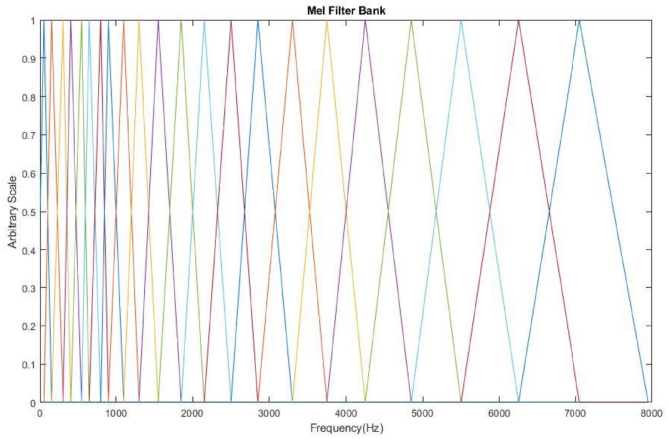


Figure 3.6: Triangular Mel Filter bank with 22 filters

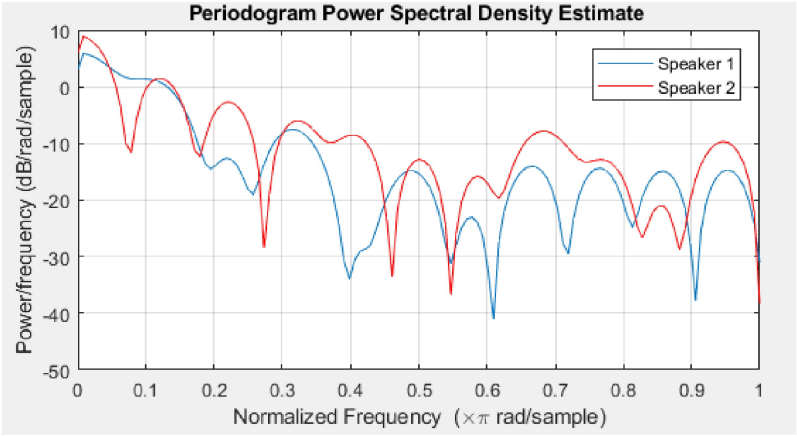


Figure 3.7: Comparing Power Spectral densities of a frame of two different speakers.

### 3.3.5 Discrete Cosine Transform

The obtained feature vectors are highly correlated, in order to de-correlate them Discrete Cosine transform (DCT) is used. This de-correlation is needed so that a naive covariance matrix can be used which is explained in section 3.4 . Generally the more overarching spectral shapes are more important than the noisy details in the spectrum. So, we take the Discrete Cosine transform (DCT) and discard the higher order coefficients and keep only the first 15 coefficients that holds the important spectral shape of a speaker. From the obtained 22 coefficients of Y(k) only 15 are considered as MFCC feature vectors and is calculated as follows,

$$b_n = \sum_{m=1}^{M} [log_{10} Y(m)] \cdot cos\left[\frac{\pi n}{M} \cdot \left(m - \frac{1}{2}\right)\right] \tag{3.14}$$

In the above equation logarithm of Y is taken to mimic the Human ear perception of loudness. The Spectogram plot of the Input signal gives an idea of the speech power level used by a

particular speaker at different frequencies as illustrated by the Figure 3.8. The coefficient $b_1$ represents the power over all frequency bands and $b_2$ represents the balance between low and high frequency components within the signal frame. The other cepstral coefficients contain the details of the spectrum shape as shown in Figure 3.9. The DCT is restricted to only 15 features as the higher order coefficients map the fast variations in speech and noise of the frame, and mapping these will reduce the accuracy of the model [1].

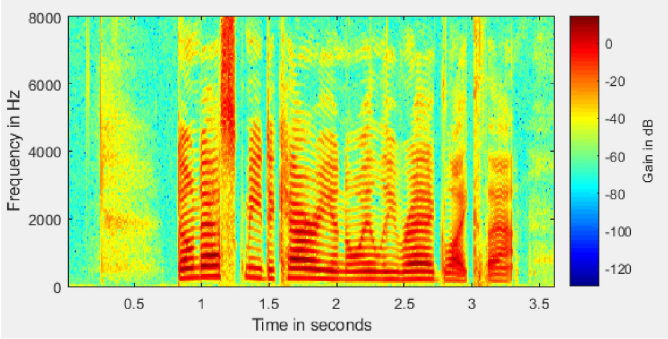


Figure 3.8: Spectogram plot of an audio file of a speaker.

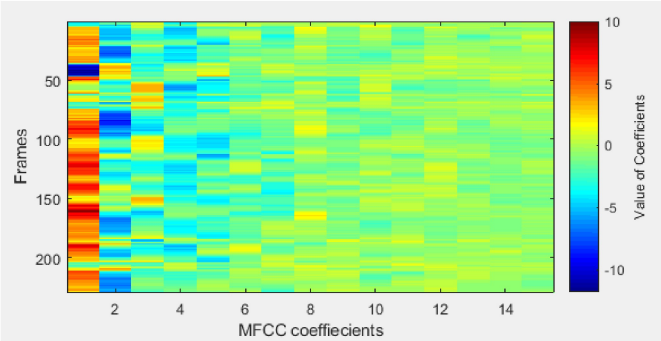


Figure 3.9: Mel Cepstral Coefficients of an audio file of a speaker.

## 3.4 Probabilistic Model of Speech

A parametric modelling technique called Guassian Mixture Model (GMM) is used to form a statistical model of the feature vectors of each speaker. The GMMs can be thought of as a generalisation of k-means where each cluster is allowed to have its own covariance matrix.

A Universal background Model (UBM) is pre-trained on a set of 462 test speakers from the TIMIT database. This model is used as a reference to limit the the training data and reduce the computational expense. The model gives a generalized mean, naive co-variance (restricted only to diagonal elements) and mixing factor (weights). It consists of K= 49 modes in 15 dimensional feature space. Here, each $k^{th}$ mode models the emotion behind the voice like happiness, sadness, anger and other Human expressions.

A general GMM $\lambda$ with K modes is described by the following probabilistic density function(pdf) :

$$p(\underline{b}|\lambda) = \sum_{k=1}^{K} w_k \cdot p(\underline{b}|\underline{\mu}_k, C_k) \tag{3.15}$$

$$p(\underline{b}|\underline{\mu}_k, C_k) = \frac{1}{(2\pi)^{\frac{D}{2}} \cdot |C_k|^{\frac{1}{2}}} \cdot e^{-\frac{1}{2}(\underline{b}-\underline{\mu}_k)^T C_k^{-1} (\underline{b}-\underline{\mu}_k)} \tag{3.16}$$

where, $\underline{b}$ is a 15 dimensional feature vector and for K= 49 modes the means $\underline{\mu}_k$ is 15 dimensional for every value of k and a naive covariance matrix for 49 modes $C_k$ is 49 x 15 in dimension. The mixing factor or weights $w_k$ has to satisfy the condition as per equation 3.17.

$$\sum_{k=1}^{K} w_k = 1 \tag{3.17}$$

The naive GMM-UBM model is given by,

$$p_{UBM}(\underline{b}) = \sum_{k=1}^{K} w_{UBM,k} \cdot p(\underline{b}|\underline{\mu}_{UBM,k}, C_{UBM,k}) \tag{3.18}$$

where $\{w_{UBM,k}, \underline{\mu}_{UBM,k}, C_{UBM,k}\}$ represent the GMM parameters of the UBM. They are estimated from the UBM training data.

### 3.4.1 Speaker Model Adaptation

Every test speaker training data $B = \{b_1, ...., b_T\}$ is adapted to the given GMM-UBM reference model from equation 3.18 as illustrated in Figure 3.10.

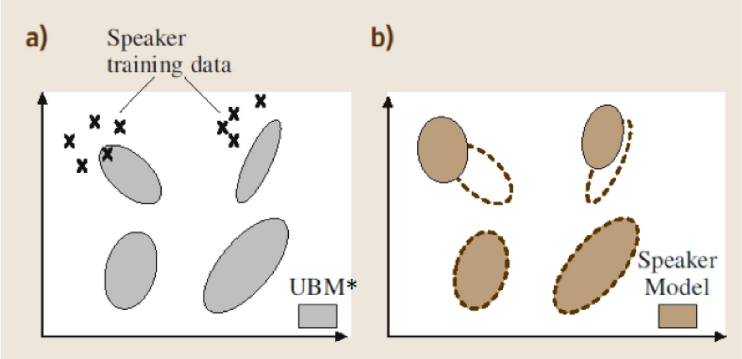


Figure 3.10: Speaker model adaptation.[7]

The adapted mean covariance and weights $\{w_k, \underline{\mu}_k, C_k\}$ for each of the speaker $\lambda$ is calculated with reference to the UBM model as follows,

$$P(k|\underline{b}_t) = \frac{w_{UBM,k} \cdot p(\underline{b}|\underline{\mu}_{UBM,k}, C_{UBM,k})}{p_{UBM}(\underline{b}_t)} \tag{3.19}$$

$$\underline{\hat{\mu}}_k = \frac{1}{\sum_{t=1}^{T} p(k|\underline{b}_t)} \cdot \sum_{t=1}^{T} p(k|\underline{b}_t) \cdot \underline{b}_t \tag{3.20}$$

$$\hat{\underline{\mathbf{C}}}_k = \left[ \frac{1}{\sum_{t=1}^{T} p(k|\underline{b}_t)} \cdot \sum_{t=1}^{T} p(k|\underline{b}_t) \cdot \underline{b}_t \cdot \underline{b}_t^{\,T} \right] - \underline{\hat{\mu}}_k \cdot \underline{\hat{\mu}}_k^{\,T} \tag{3.21}$$

$$\underline{\hat{w}}_k = \frac{1}{T} \cdot \sum_{t=1}^{T} p(k|\underline{b}_t) \quad where, \sum_{k=1}^{K} \underline{\hat{w}}_k = 1 \tag{3.22}$$

The obtained parameters are combined with the k-th mode UBM model as follows,

$$\underline{\mu}_k = \alpha_k \cdot \underline{\hat{\mu}}_k + (1 - \alpha_k) \cdot \underline{\mu}_{UBM,k} \tag{3.23}$$

$$\underline{\mathbf{C}}_k = \alpha_k \cdot \hat{\underline{\mathbf{C}}}_k + (1 - \alpha_k) \cdot \underline{\mathbf{C}}_{UBM,k} \tag{3.24}$$

$$\underline{w}_k = \alpha_k \cdot \underline{\hat{w}}_k + (1 - \alpha_k) \cdot \underline{w}_{UBM,k} \tag{3.25}$$

with,

$$\alpha_k = \frac{\sum_{t=1}^{T} p(k|\underline{b}_t)}{\gamma + \sum_{t=1}^{T} p(k|\underline{b}_t)} \tag{3.26}$$

The relevance factor $\gamma = 0.1$ is chosen based on trial and error to get the best detection rate possible. It is observed that if the GMM parameters are more speaker dependent than UBM, then the model is more accurate and vice versa.

To better understand the adapted models, we visualized few speaker models in 2-D. First Principal Component Analysis (PCA) of covariance matrix of individual speaker data was carried out [8]. PCA gives us eigen basis for individual speaker data. Eigen vectors corresponding to two highest eigen values were used as basis vectors and the model mean is transformed from 15 dimensions to 2 dimensions. In order to plot all the gaussian modes for a speaker model, we additionally need covariance matrices in 2 dimensions. As we have considered the covariance matrix in 15 dimensions to be naive, we choose two largest elements from diagonal and take them as diagonal elements in transformed 2*2 covariance matrix. While plotting the gaussian modes, these diagonal elements become the axes of the ellipses in 2 dimensions to represent variance of the gaussian.

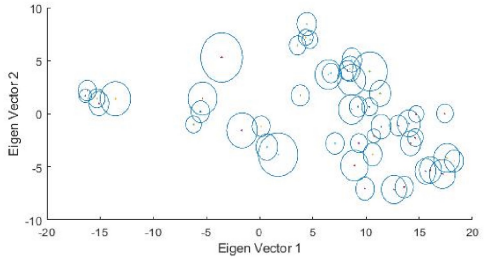


Figure 3.11: UBM Model

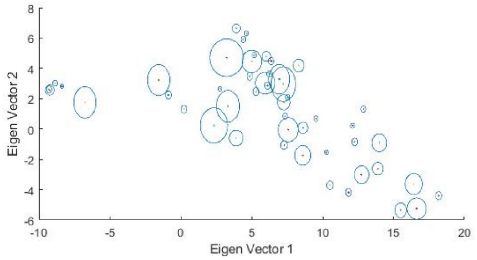


Figure 3.12: Female speaker Model

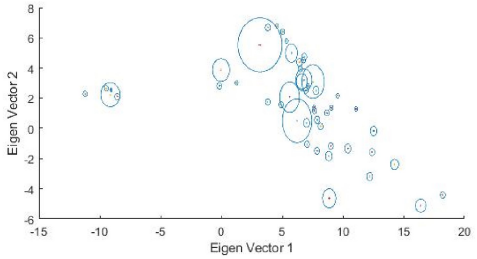


Figure 3.13: Male speaker Model

From the Figures 3.11, 3.12 and 3.13 , we can see that UBM gaussian modes have high variance in many modes as it is a generic representation of human voice spectrum. For a female and male speaker, the modes have different distribution highlighting the differences

in the nature of voices.

## 3.5 Speaker Identification

The frames of test speaker's one audio file is passed through all the trained GMM models from the section 3.4. We find the utterance score $S(Y|\lambda)$ for all the speaker models. The highest utterance score's argument is the most likely speaker the model detects based on the given input test frames as per the following,

$$S(Y|\lambda) = \sum_{t=1}^{T} log\ p(\underline{b_t}|\lambda) \tag{3.27}$$

The estimated speaker is as per the formula,

$$\hat{s} = arg\ \ max_{\lambda}(log\ S(Y|\lambda)) \tag{3.28}$$

Based on the experimental results, it was observed that there should be atleast 100 frames available for the test audio file for better accuracy. when the frames are less than 100, the GMM model based approach behaves poorly with just 70% accuracy.

## 3.6 Results and Conclusion

Cross validation with 10 iterations is performed in a way that at a time one of the 10 audio files is considered as test file and the rest as training files from TIMIT database(including our recorded voices) . Confusion matrix of all the iterations is summed and illustrated in the Figure 3.14.

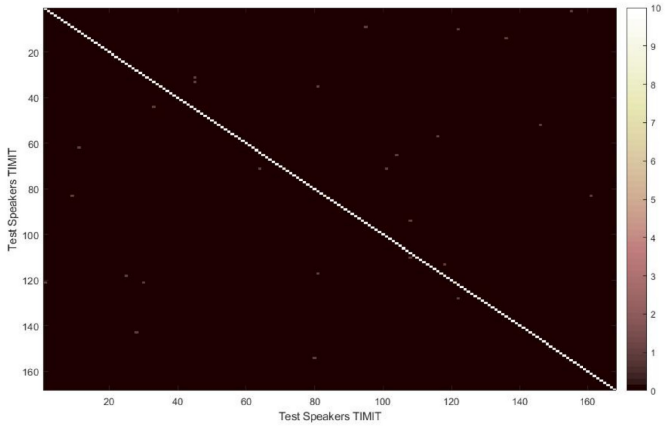


Figure 3.14: Confusion Matrix for 10 iterations

The Table 3.1 below indicates the accuracy obtained at each iteration.

Table 3.1: Cross validation accuracy

| Iterations | Accuracy (%) |
|---|---|
| 1 | 100 |
| 2 | 99.41 |
| 3 | 98.21 |
| 4 | 97.62 |
| 5 | 97.02 |
| 6 | 99.40 |
| 7 | 98.80 |
| 8 | 97.02 |
| 9 | 98.21 |
| 10 | 100 |

It can be seen that the model predicts accurately with an average accuracy of 98%. The detection accuracy rate can be further increased if we add extra features like pitch, zero cross rate, higher order - spectrum momentum along with the MFCC features.

The solution implemented applies only when the the test frames are from a known speaker. The problem of detecting an unknown speaker may be solved by the finding the utterance scores of an unknown speaker audio file passed through the GMM-UBM model and the known speaker models separately. Later, we perform a score normalization as illustrated in the figure below to identify if the speaker is unknown.

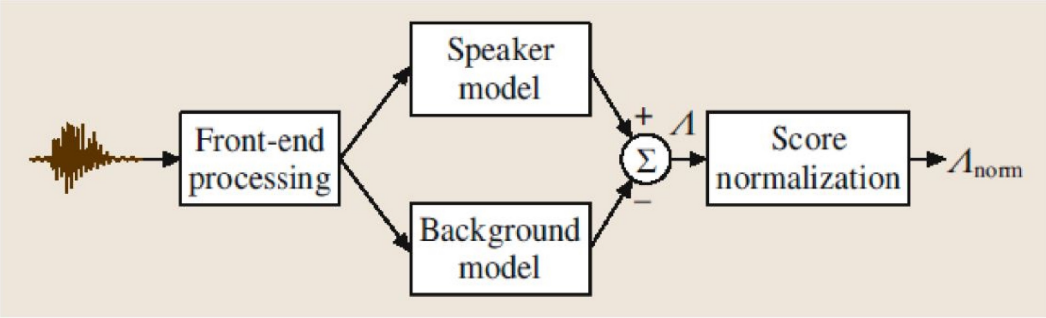


Figure 3.15: Identifying an unknown speaker [7]

# Bibliography


[1] Douglas A. Reynolds, Robert B. D. Thomas F. Quatieri Q. Thomas F. Quatieri: *Speaker Verification Using Adapted Gaussian Mixture Models.* In: Digital Singal Processing 10 (2000).
*http* : *//speech.csie.ntu.edu.tw/previous*$_v$*ersion/S peaker%*20*Verification%*20*Using%*20*Adapted%*20

[2] Kinnunen, H. L. T.: *An overview of text-independent speaker Recognition: form features to supervectors.*Version: 2009
*http* : *//cs.joensuu.fi/pages/tkinnu/webpage/pdf/speaker*$_r$*ecognition*$_o$*verview.pdf*

[3] Weber-Fechner law.: Wikipedia.
*https* : *//en.wikipedia.org/wiki/Weber*˘*Fechner*$_l$*aw*

[4] Hoang Do, Alex A. Ivan Tashev T. Ivan Tashev: *A new speaker identification algorithm for gaming scenarios.*(2001)

[5] Automatic speech recognition *Feature Extraction: MFCC vectors.*
*http* : *//www*2*.cmpe.boun.edu.tr/courses/cmpe*362*/spring*2014*/files/projects/MFCC%*20*Feature%*

[6] SR Wiki *Mel-Frequency Cepstral Coefficients.*
*http* : *//recognize* − *speech.com/feature* − *extraction/mfcc*

[7] Reynolds and Campbell.: *Speaker Recognition.*2008, in Benesty et al., (Eds)
*http* : *//research.cs.tamu.edu/prism/lectures/sp/l*16*.pdf*

[8] Herve Abdi and Lynne J. Williams: *Principal component analysis:*2008, in Benesty et al., (Eds)
*https* : *//www.utdallas.edu/ herve/abdi* − *awPCA*2010*.pdf*

[9] Yang, B. : *Detection and pattern recognition: Lecture notes and video recordings. University of Stuttgart,2016.*

[10] Institute für Signalverarbeitung und Systemtheorie: *PÜL Statistical Signal Processing Pattern Recognition.*

[11] Almeida,M.Barros d.; Padua Braga,A de; Braga,J.P: *SVM-KM: speeding SVMs learning with a priori cluster selection and k -means*2000, in Neural Networks Proceedings

# Declaration

Herewith, we declare that we have developed and written the enclosed thesis entirely by ourself and that we have not used sources or means except those declared.

This thesis has not been submitted to any other authority to achieve an academic grading and has not been published elsewhere.