3/3/2021

DSP Simulation Assignment 2

Speaker Identification and Verification using Spectral Energy Characteristics

Problem Statement: -

- Obtain Speech recordings of a combination lock number (2-4-3-9-1-7-5) uttered by different persons at different times of the day.
- Observe visually identifiable similarities/dissimilarities among the spectrums of different groups/subgroups.
- Filter the signals using multiple equally spaced filters and observe similarity in terms of Euclidean distance.
- Observe the effect of number or uniformity of band pass filter banks on similarity heatmap.
- Find a suitable algorithm that that allows only a particular speaker to be able to open the lock by uttering the combination lock number. Find the false acceptance / false rejection rate of the algorithm.
- Extract the pitch of the speech signal following a suitable published literature.

- Rudrajyoti Roy (18EC10047)
- Swarnava Sarkar (18EC10070)
- Aishik Mandal (18EC10074)
- Debjoy Saha (18EC30010)

OVERVIEW: -

Our entire assignment revolves around the two following concepts, namely speaker verification and speaker identification.

There are two major applications of speaker recognition technologies and methodologies. If the speaker claims to be of a certain identity and the voice is used to verify this claim, this is called ***verification*** or ***authentication***. On the other hand, ***identification*** is the task of determining an unknown speaker's identity. In a sense, speaker verification is a 1:1 match where one speaker's voice is matched to a particular template whereas speaker identification is a 1:N match where the voice is compared against multiple templates. From a ***security perspective***, identification is different from verification. Speaker verification is usually employed as a "gatekeeper" in order to provide access to a secure system. These systems operate with the users' knowledge and typically require their cooperation. Speaker identification systems can also be implemented covertly without the user's knowledge to identify talkers in a discussion, alert automated systems of speaker changes, check if a user is already enrolled in a system, etc. In forensic applications, it is common to first perform a speaker identification process to create a list of "best matches" and then perform a series of verification processes to determine a conclusive match. Working to match the samples from the speaker to the list of best matches helps figure out if they are the same person based on the amount of similarities or differences.

In this assignment, although the applicability of speaker identification is subtle as the number of registered users is very less than what is done/expected in real life applications, we do try to perform a small-scale speaker verification algorithm from a ***signal processing*** point of view and further identification using ***machine learning***.

Using the concepts of filters, we can design FIR filters that are equally spaced over the entire speech spectrum, with the first one being a Low Pass Filter, the last one being a High Pass Filter and all intermediate ones being Band Pass Filters. Through filtering, we can obtain energies for the corresponding segments of our speech signal. These energy components hold a lot of information about our speech signal. Thus, each speech signal is mapped to a vector of dimension 1xN, where N is the number of filters used. All these vectors can then be considered to be as simple geometrical points in a N-dimensional space. A dimensionality reduction is necessary to get the points mapped to a 2-D or 3-D space optimally without loss of too much information. Then, clustering can be performed on the basis of user labels. Now, if a random user speech signal is obtained, we can perform the above technique and allow access to the system if and only if the new user’s speech falls into one of the clusters belonging to the registered users. However, the speech signal from the same authorized user may vary for a number of reasons. In order to handle this, a threshold of similarity should be considered.

Now, it is expected that the geometry of the clusters will change with the number of filters that are used and also the spacing of the filters. Different cases of each should be taken in order to find the one that gives us a result that has a good verification or recognition.

THEORY: -

We pass the recorded audio signal through a low pass filter, some bandpass filters with no overlap in bands and a high pass filter. The center points of the filters were distributed from 0 to F_{MAX} which is 4000Hz in our case. A low pass filter (LPF) is a filter that passes signals with a frequency lower than a selected cutoff frequency and attenuates signals with frequencies higher than the cutoff frequency. Similarly, a high pass filter passes signals above cutoff frequency and attenuates the rest. Bandpass Filter has a lower cutoff and a higher cutoff. It passes signals within the band from lower cutoff to higher cutoff and attenuate the rest. Butterworth Filter was used in our case as LPF, HPF and BPF. The Butterworth filter is a type of signal processing filter designed to have a frequency response as flat as possible in the passband. It is also referred to as a maximally flat magnitude filter. We have used a Butterworth filter of order 5.

We associate a vector of size equal to the number of filters, to each of our samples. We pass the samples through each of the filters and get the energy of each output using the following formula:

$$Energy = \sum_{n=-\infty}^{\infty} |x[n]|^2$$

We **normalize** the Energies for each sample by dividing them by max energy output by a filter for that sample. This ensures there are no scaling issues.

First, we try to find similarity and dissimilarity among various samples by the means of **Euclidean Distance**. The Euclidean Distance between two vectors x1 and x2 of size n is:

$$Dist = \sqrt{\sum_{i=0}^{n}(x1[i] - x2[i])^2}$$

But we do not get satisfactory results and thus move towards classification algorithms for the same. We use K-NN Classification with the 32 Data Points divided into 28 for training and 4 for validation. It assigns the given datapoint a class which the majority of its k-nearest Neighbors has. In our case k=5 and thus we check the classes of the 5 nearest datapoint and assign a class which is in majority. Since the Vectors are multi-dimensional, we need to do dimension reduction to 2 for visualization. We use **t-Distributed Stochastic Neighbor Embedding(t-SNE)** for the same. It is different from PCA as PCA is a linear reduction technique whereas t-SNE is an unsupervised non-linear technique. The t-SNE algorithm calculates a similarity measure between pairs of instances in the high dimensional space and in the low dimensional space. It then tries to optimize these two similarity measures using a cost function.

2-D `Meshgrid` was used to classify all points on the R^2-space in order to get the decision boundaries.

KNN for Speaker Identification and Verification

AS earlier mentioned, we are using the K-nearest neighbour algorithm for classification of each voice signal.

K-NN algorithm: In statistics, the k-nearest neighbour algorithm (k-NN) is a non-parametric supervised classification method. In k-NN classification, the output is a class membership. An object is classified by a plurality vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbor. In our case, the energy vectors (filter-bank outputs) are used as features for k-NN algorithms.

Speaker Identification: The primary task of a speaker identification system is to classify each incoming voice sample as belonging to any one of the reference speakers. For achieving this task, we first divide the voice dataset into separate train and validation splits (ensuring that the validation split consists of at least one speech sample from each reference speaker). Then, we train the k-NN algorithm on the training split and view the model performance on the validation split of the dataset. Model performance report provided in discussion.

Speaker Verification: In speaker verification, the task is to classify each voice as to whether it belongs to a prespecified set of speakers or not. So, in addition to classifying the incoming voice sample to one of the speakers, an additional task is to detect when the speech sample belongs to an entirely new speaker not belonging to the reference set. Now, the k-NN algorithm is not suited for this task, since it maps the entire feature space to some speaker and provides no provision to detect outliers. Modern speech verification algorithms employ techniques like contrastive learning, which are quite data-intensive and hence, not suited to our task.

Instead, we use a modified version of the k-NN. In this algorithm, after the classification of the incoming speech sample, a deviation measure is calculated using the reference speech vector of the classified speaker, where the reference speech vector is taken as the mean of speech vectors for each registered speaker. If the deviation is found to exceed a user-defined threshold value, the speaker is classified as an unregistered/new speaker.

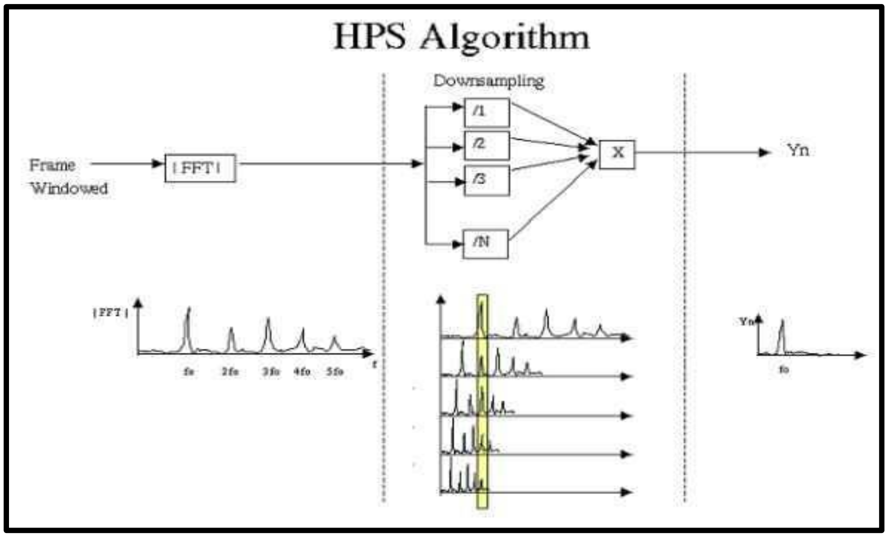
The threshold value is a user-defined hyperparameter and can be tuned to achieve desirable false acceptance and false rejection rates (Lower value of either one can be more desirable based on the target application). For robust performance, we take the speaker-dependent threshold as the variance in the speech vectors for that speaker multiplied by a tunable scaling factor. This approach normalizes the inherent variations in the speech signals for each person.

Pitch Detection :-

Pitch :- Speech can be broadly categorized as ***voiced*** and ***unvoiced*****.** In the case of voiced speech, air from the lungs is modulated by vocal cords and results in a quasi-periodic excitation. The resulting sound is dominated by a relatively low-frequency oscillation, referred to as ***pitch***. In the case of unvoiced speech, air from the lungs passes through a constriction in the vocal tract and becomes a turbulent, noise-like excitation. In the source-filter model of speech, the excitation is referred to as the source, and the vocal tract is referred to as the filter. Characterizing the source is an important part of characterizing the speech system.

Pitch Detection Algorithm :- A ***pitch detection algorithm*** *(**PDA**)* is an algorithm designed to estimate the pitch or fundamental frequency of a quasiperiodic or oscillating signal, usually a digital recording of speech or a musical note or tone. This can be done in the time domain, the frequency domain or both. In the following section we discuss one such algorithm.

Harmonic Product Spectrum Technique :- If the input signal is a speech signal, then its spectrum should consist of a series of peaks, corresponding to fundamental frequency with harmonic components at integer multiples of the fundamental frequency. Hence when we compress the spectrum a number of times (downsampling), and compare it with the original spectrum, the strongest harmonic peaks line up. The first peak in the original spectrum coincides with the second peak in the spectrum compressed by a factor of two, which coincides with the third peak in the spectrum compressed by a factor of three. Hence, when the various spectrums are multiplied together, the result will form a clear peak at the fundamental frequency. The overview of the whole technique can be understood from the following diagram.



First, we divide the input signal into segments by applying a **Hamming window**, where the window size and overlap size are given as an input. For each window, we utilize the Short-Time Fourier Transform to convert the input signal from the time domain to the frequency domain. Once the input is in the frequency domain, we apply the Harmonic Product Spectrum technique to each window.

The HPS algorithm involves two steps: **downsampling and multiplication**. To downsample, we compressed the spectrum twice in each window by resampling: the first time, we compress the original spectrum by two and the second time, by three. Once this is completed, we multiply the three spectra together and find the frequency that corresponds to the peak (maximum value). This particular frequency represents the fundamental frequency of that particular window.

The following research articles are referred for understanding and implementing the pitch detection algorithm.

[1] L. Rabiner, "On the use of autocorrelation analysis for pitch detection," in *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 25, no. 1, pp. 24-33, February 1977, doi: 10.1109/TASSP.1977.1162905.

[2] Hess W.J. (1982) Algorithms and Devices for Pitch Determination of Speech Signals. In: Haton JP. (eds) Automatic Speech Analysis and Recognition. NATO Advanced Study Institutes Series (Series C — Mathematical and Physical Sciences), vol 88. Springer, Dordrecht. https://doi.org/10.1007/978-94-009-7879-9_3

[3] Cuadra, P. et al. “Efficient Pitch Detection Techniques for Interactive Music.” *ICMC* (2001).

CODE AND OBSERVATIONS: -

- The voice samples are recorded using Audacity (Freeware). The project is set to 8 kHz Mono channel and saved using 16-bit .WAV uncompressed encoding scheme. The repository of voice samples is distributed to our class here.
- Additionally, some test samples are provided by our group members to test the KNN model and detect intruder. Our idea is that the combination lock would only open for those voices on which our algorithm is trained. The working directory is included here.
- The approach is elaborated in the MATLAB live scripts which are attached with this report as **Code_1.pdf (Euclidean Distance and pitch detection)** and **Code_2.pdf (KNN based classification and verification)** respectively.

Female voice sample - #1: Nivedita Majee (18EC30027)

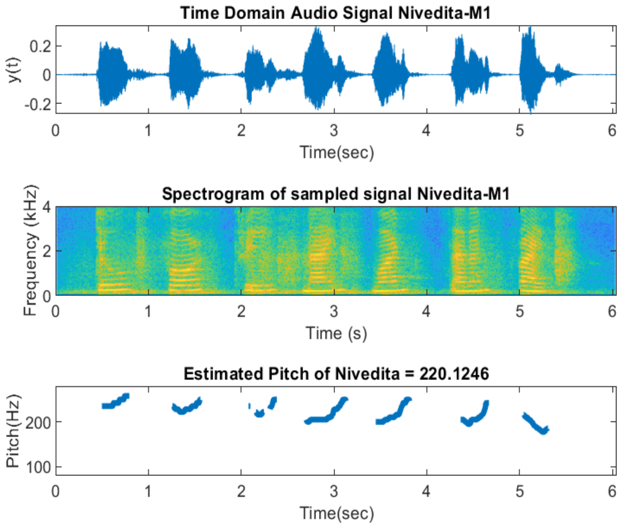


Fig: Speech Characteristics of Current Sample

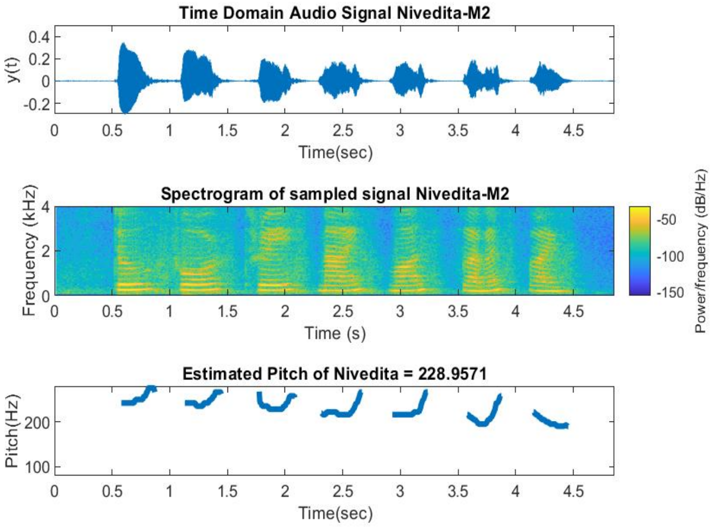


Fig: Speech Characteristics of Current Sample

Morning Samples (Average Pitch: - 225 Hz)

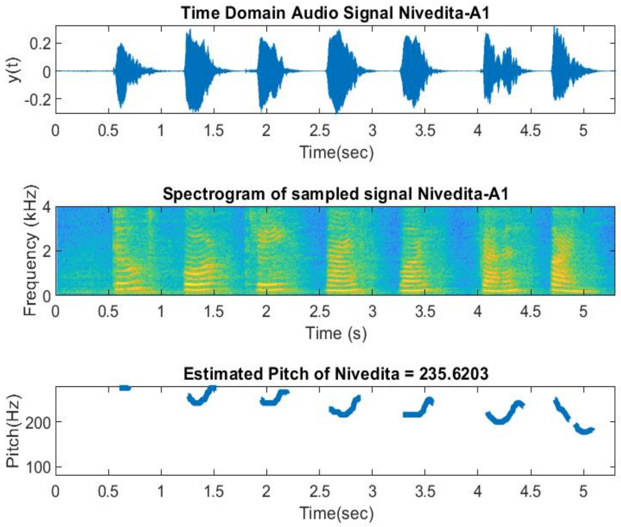


Fig: Speech Characteristics of Current Sample

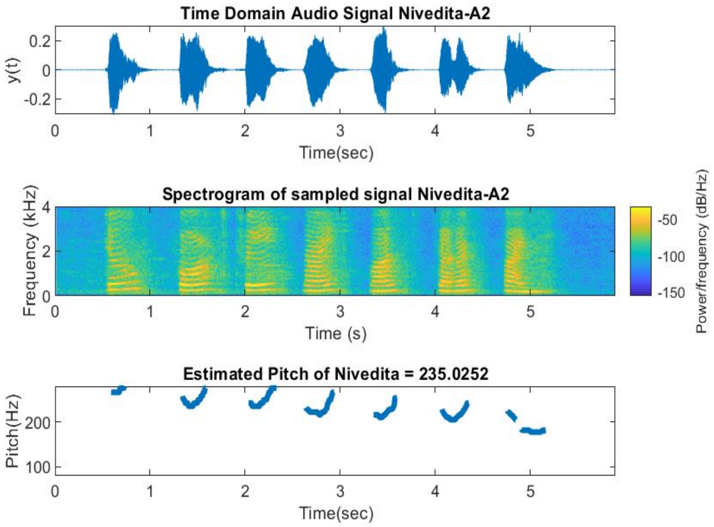
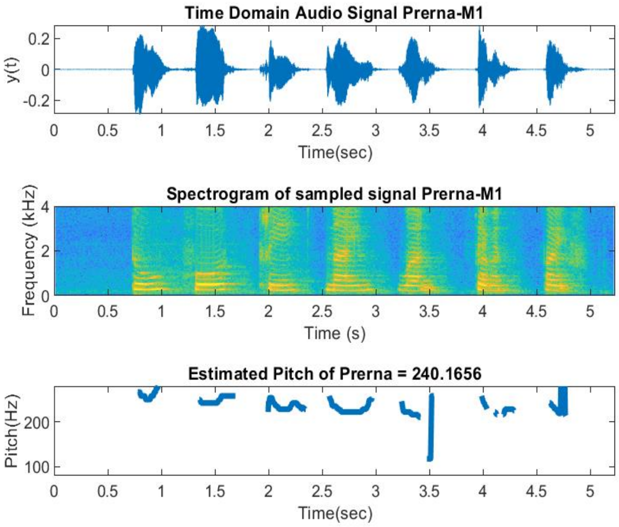


Fig: Speech Characteristics of Current Sample

Afternoon Samples (Average Pitch: - 235 Hz)

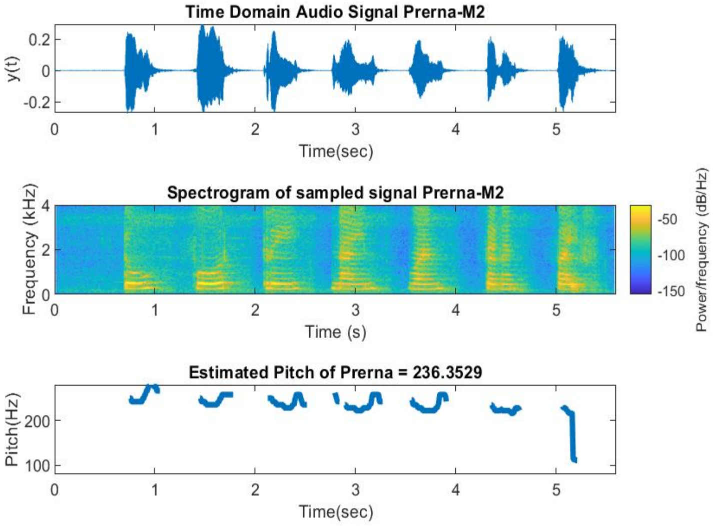
Female voice sample - #2: Prerna Goel (18EC30042)


Time Domain Audio Signal Prerna-M1

Spectrogram of sampled signal Prerna-M1

Estimated Pitch of Prerna = 240.1656

Fig: Speech Characteristics of Current Sample

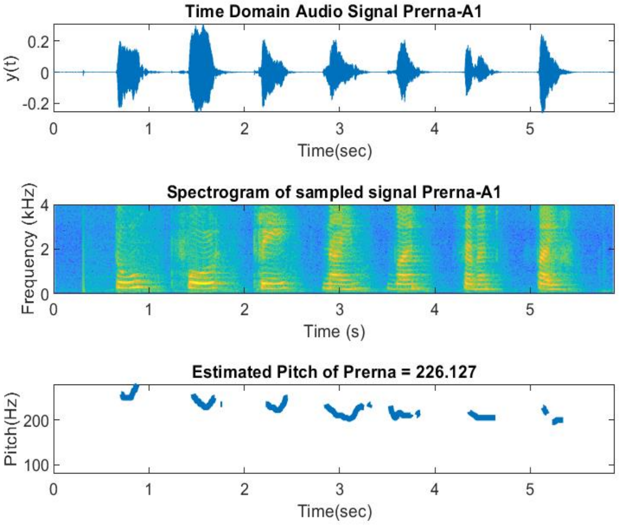

Time Domain Audio Signal Prerna-M2

Spectrogram of sampled signal Prerna-M2

Estimated Pitch of Prerna = 236.3529

Fig: Speech Characteristics of Current Sample

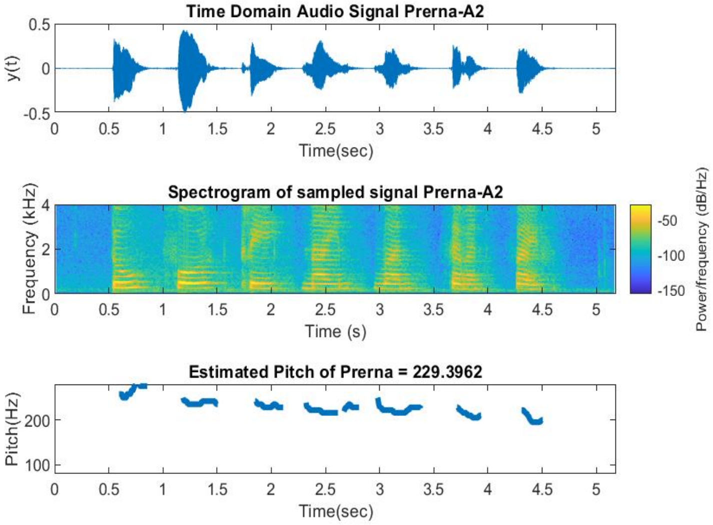
Morning Samples (Average Pitch: - 238 Hz)


Time Domain Audio Signal Prerna-A1

Spectrogram of sampled signal Prerna-A1

Estimated Pitch of Prerna = 226.127

Fig: Speech Characteristics of Current Sample


Time Domain Audio Signal Prerna-A2

Spectrogram of sampled signal Prerna-A2

Estimated Pitch of Prerna = 229.3962

Fig: Speech Characteristics of Current Sample

Afternoon Samples (Average Pitch: - 228 Hz)

Male voice sample - #1: Rudrajyoti Roy (18EC10047)

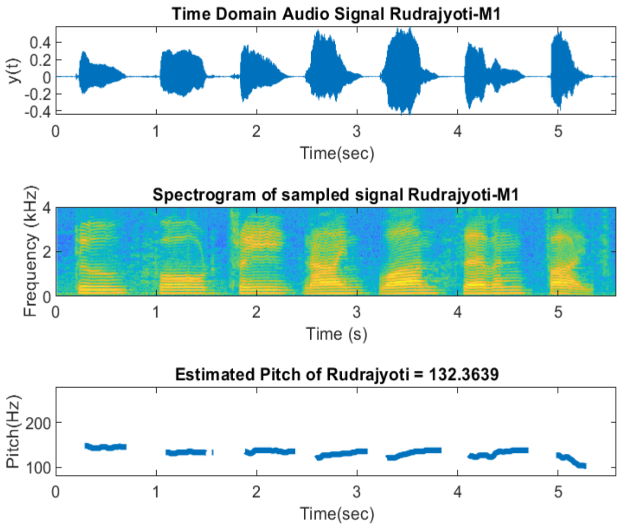


Fig: Speech Characteristics of Current Sample

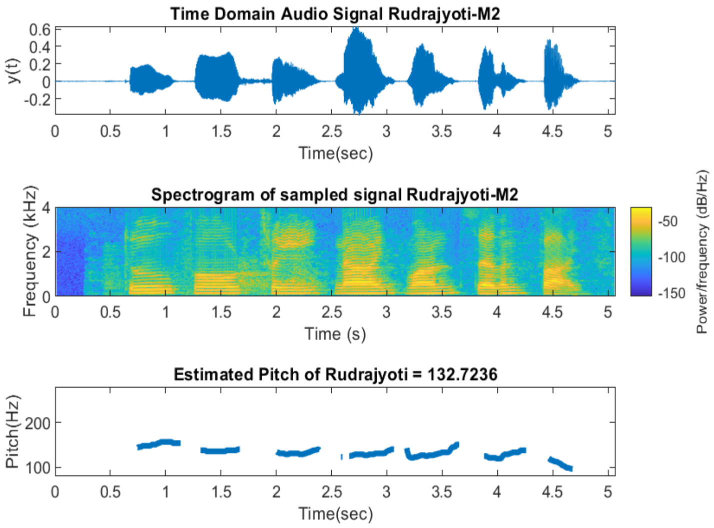


Fig: Speech Characteristics of Current Sample

Morning Samples (Average Pitch: - 132.5 Hz)

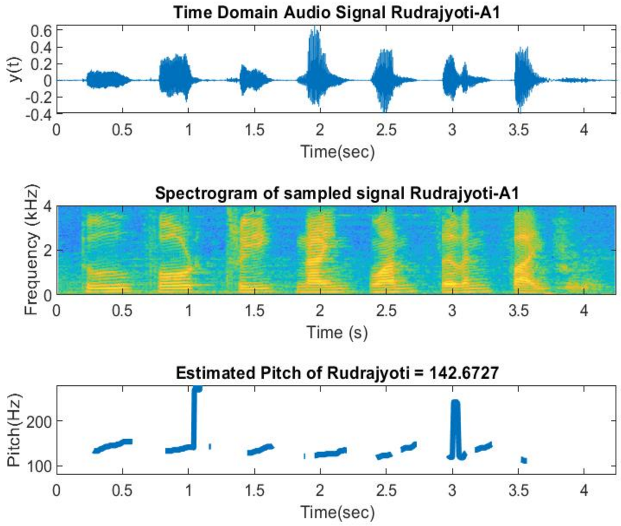


Fig: Speech Characteristics of Current Sample

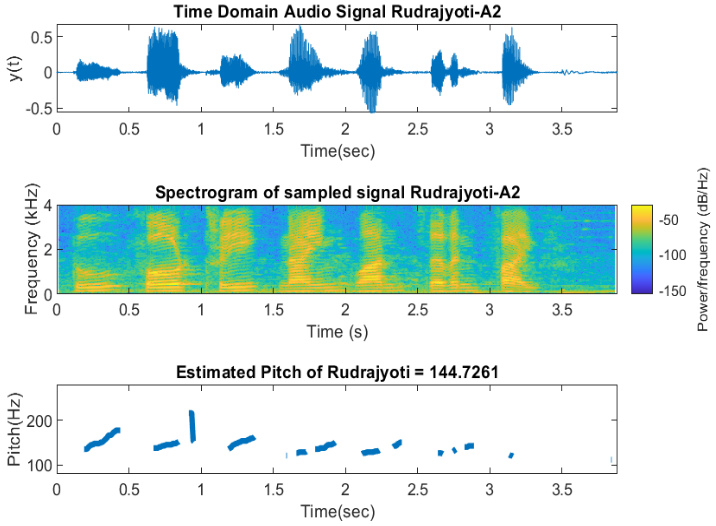


Fig: Speech Characteristics of Current Sample

Afternoon Samples (Average Pitch: - 143 Hz)

Male voice sample - #2: Swarnava Sarkar (18EC10070)

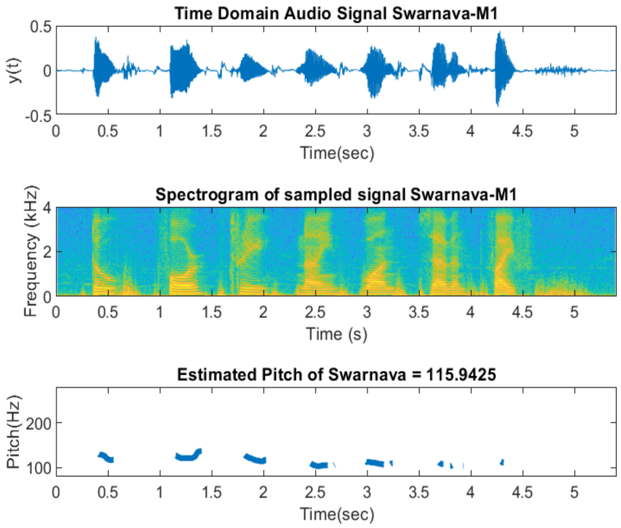


Fig: Speech Characteristics of Current Sample

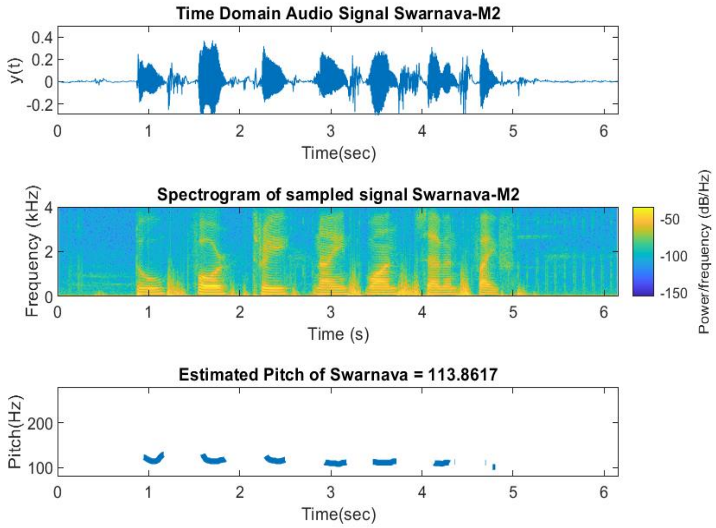


Fig: Speech Characteristics of Current Sample

Morning Samples (Average Pitch: - 114 Hz)

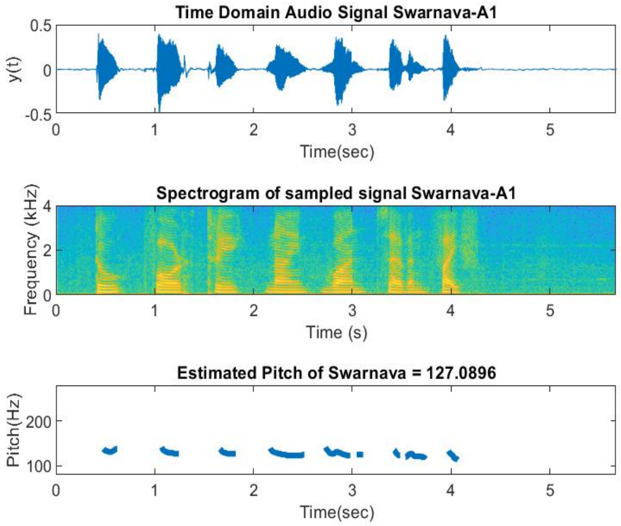


Fig: Speech Characteristics of Current Sample

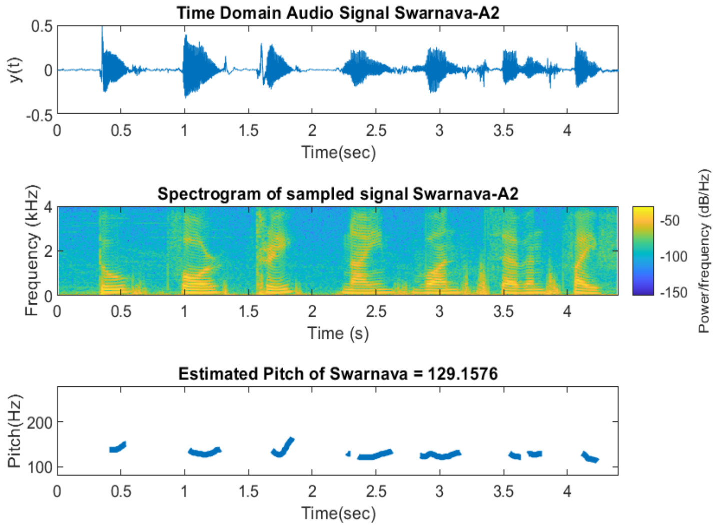
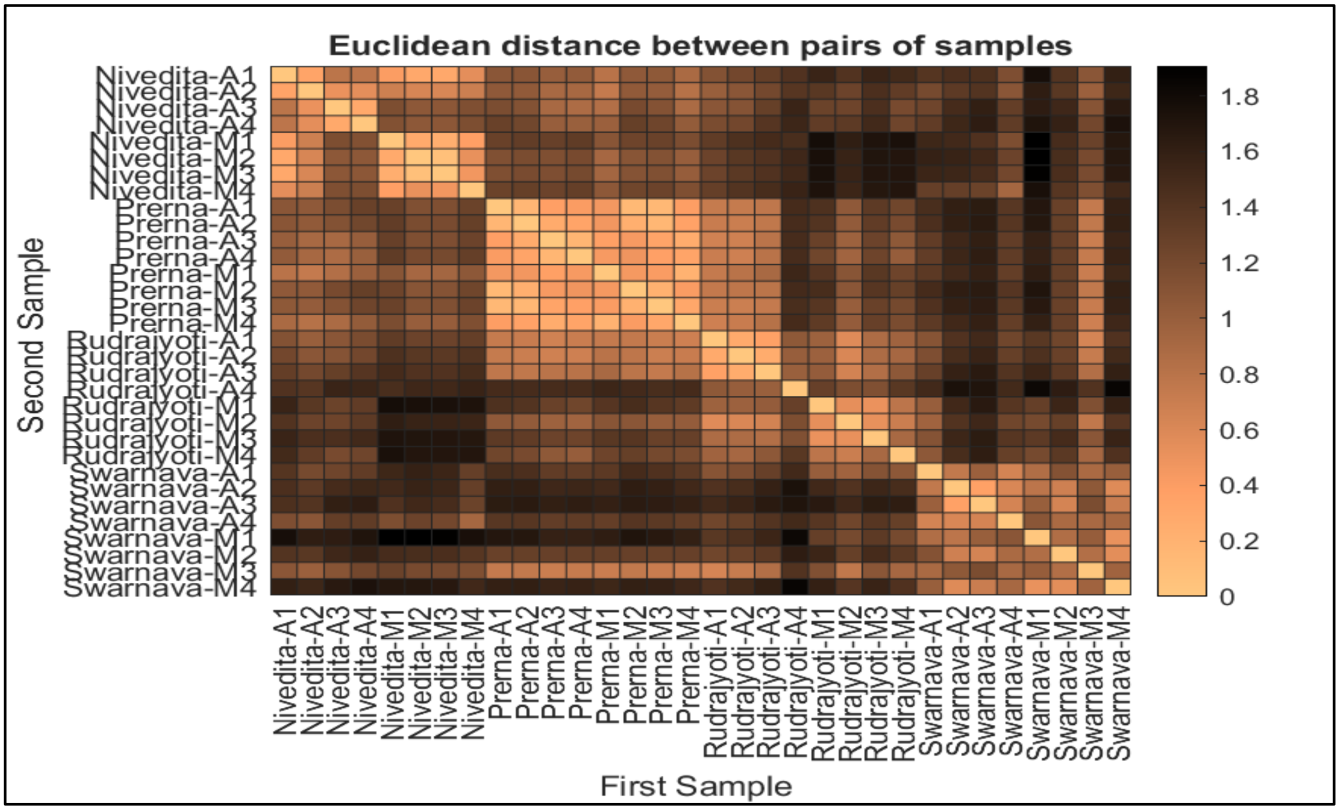
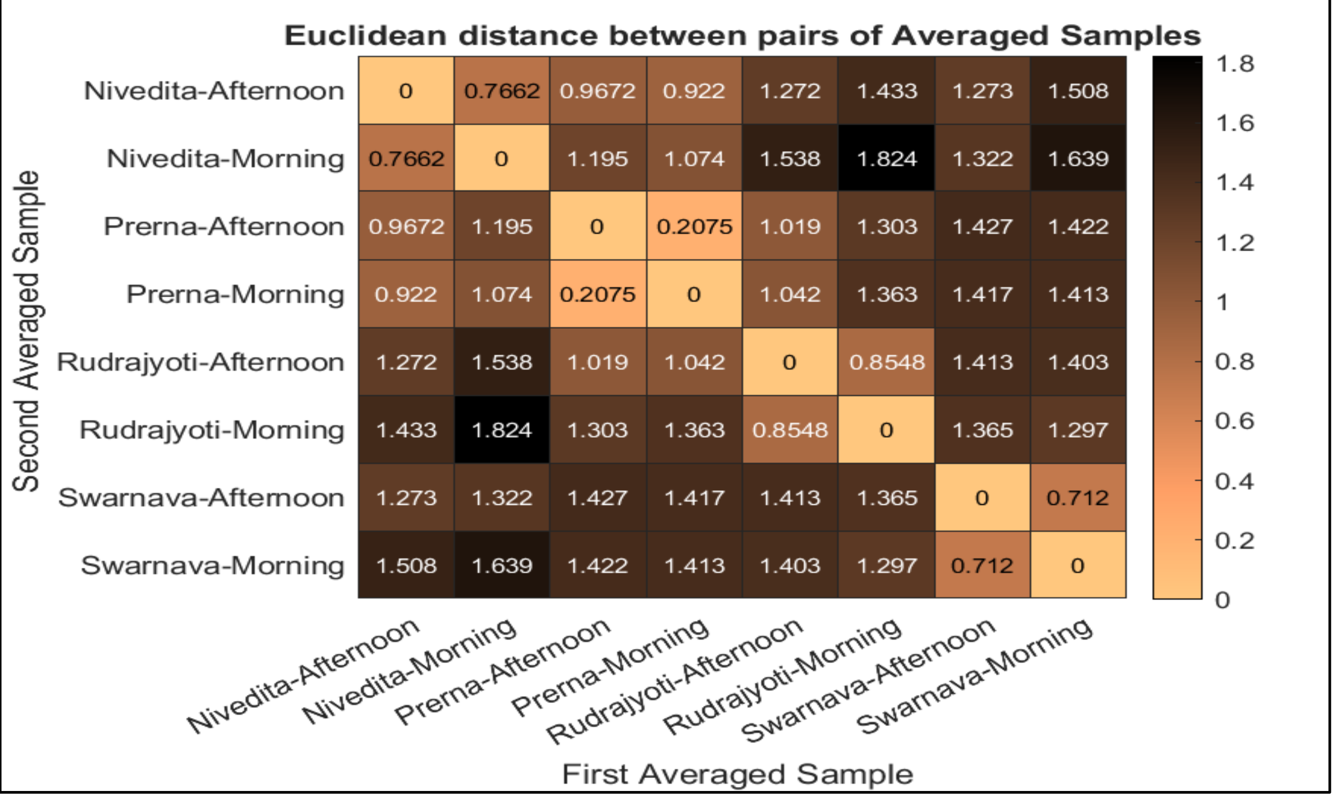


Fig: Speech Characteristics of Current Sample

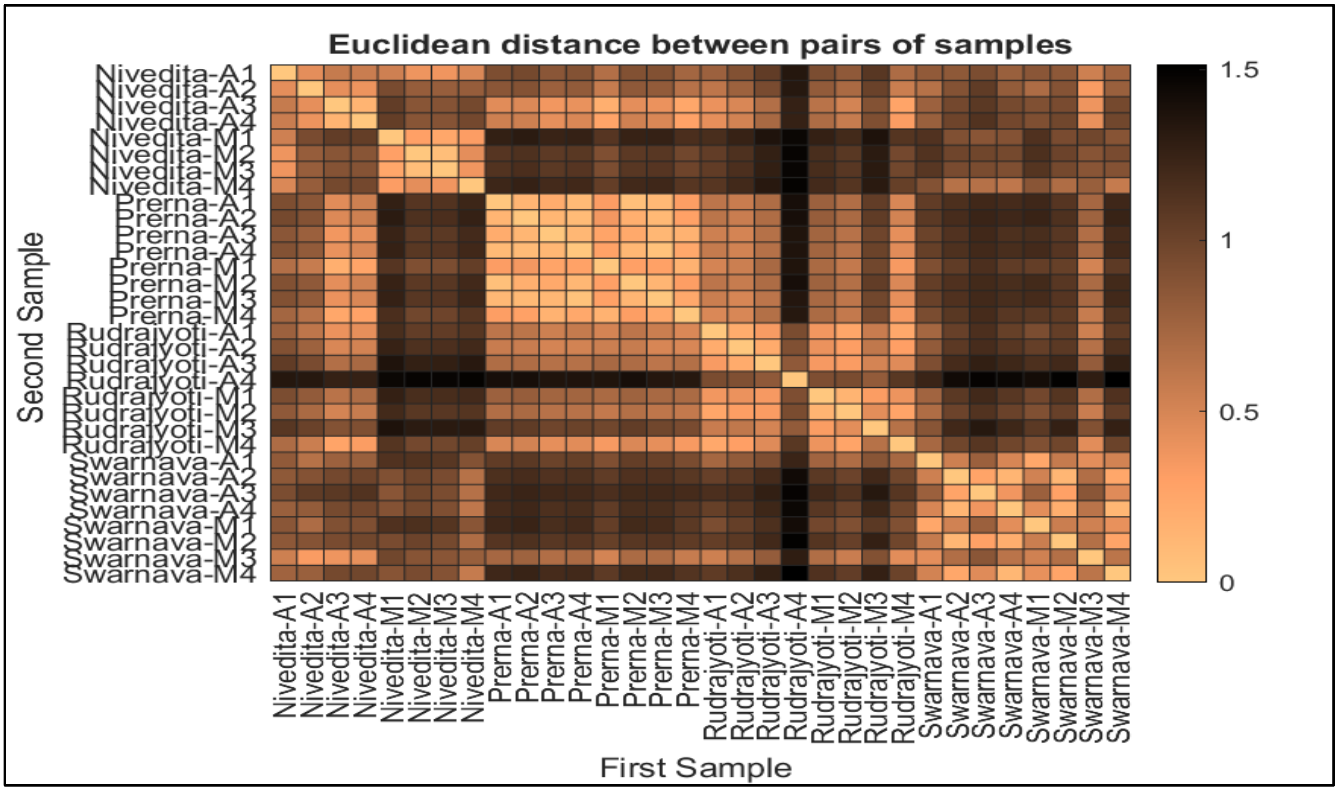
Afternoon Samples (Average Pitch: - 128 Hz)

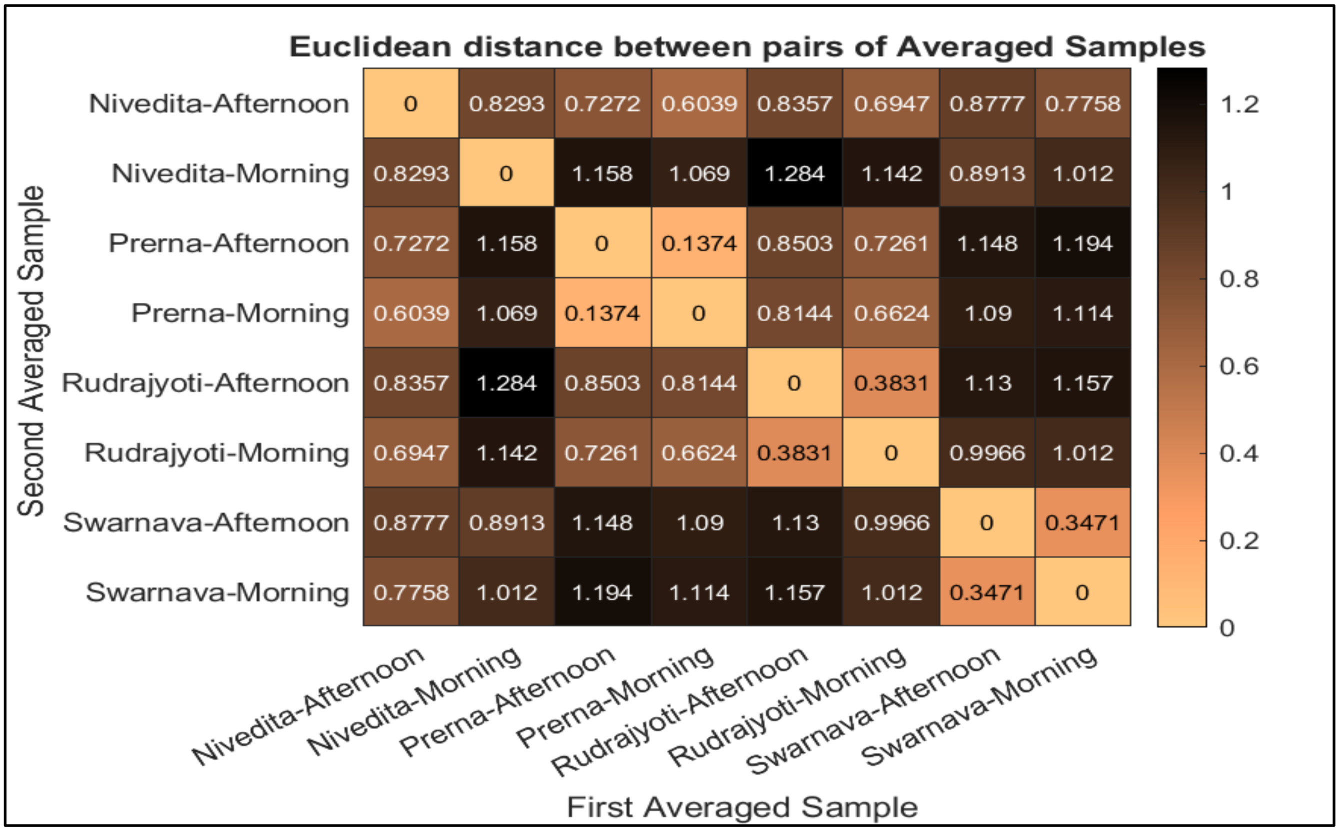
Heatmap - Non-Uniform Filter Placement





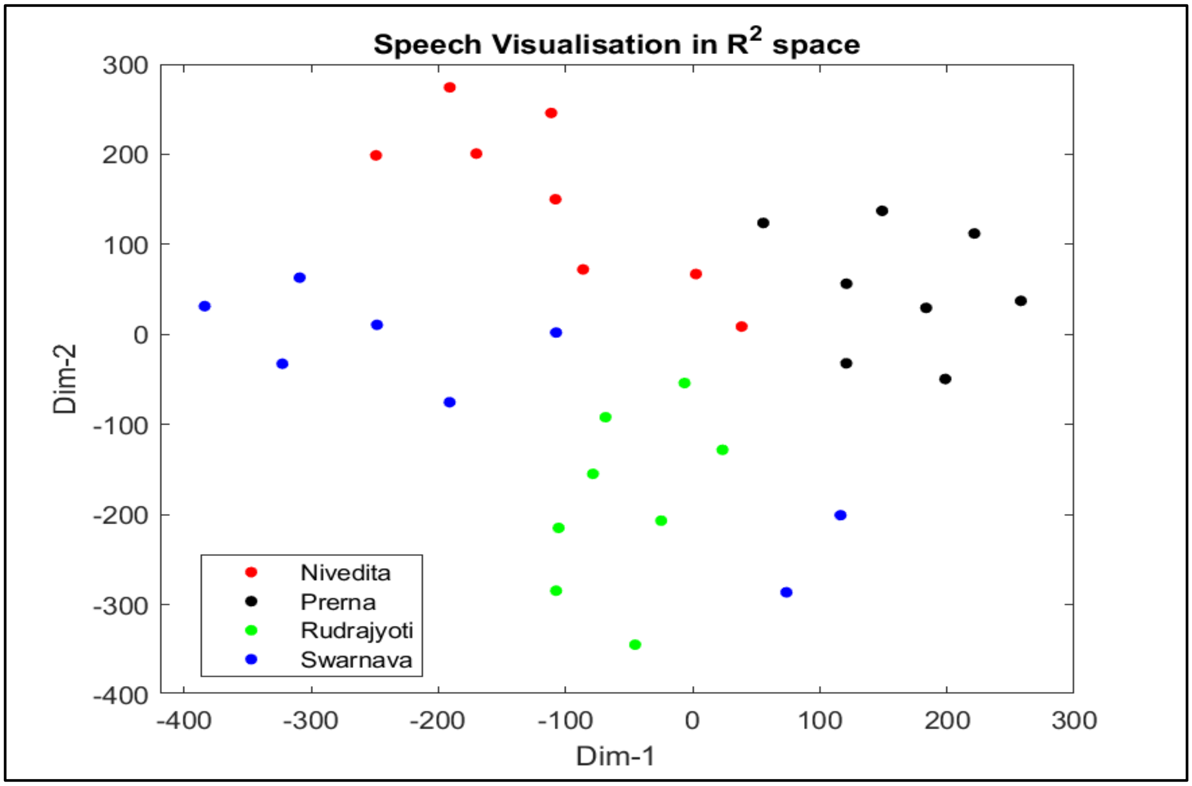
Heatmap - Uniform Filter Placement

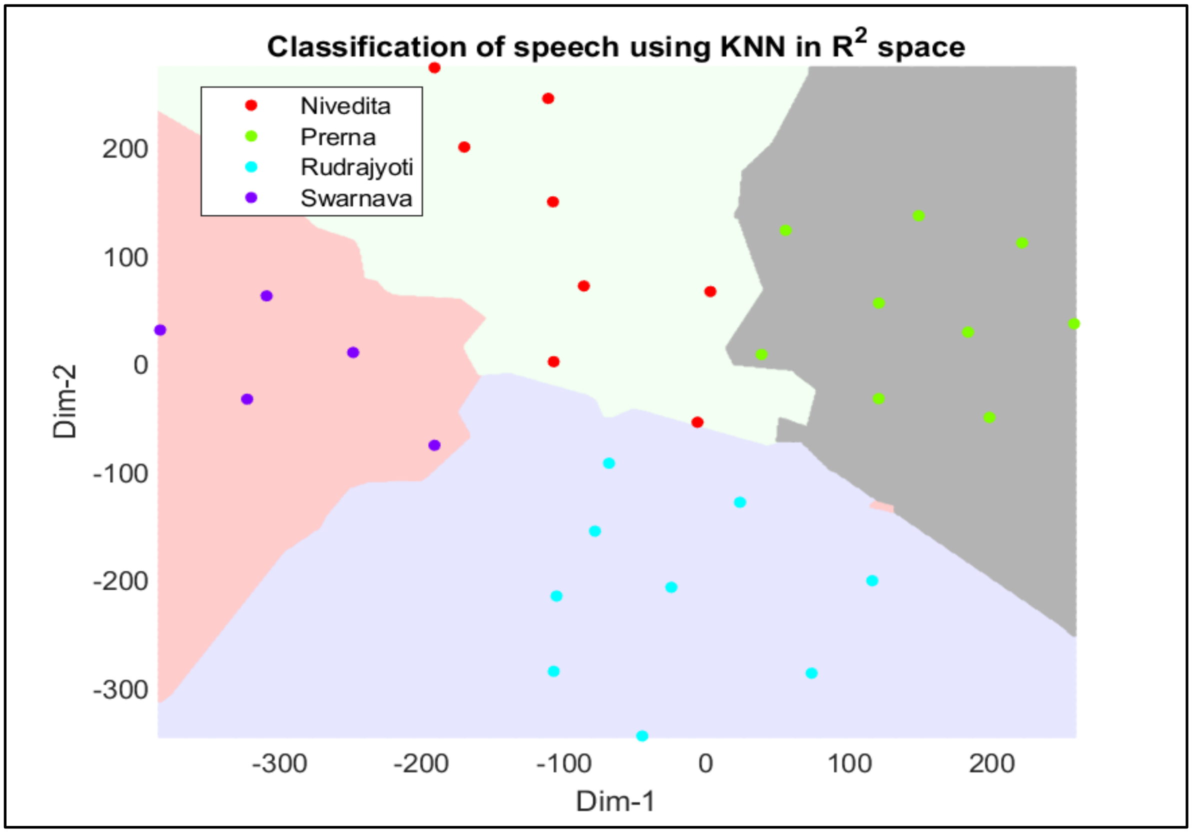




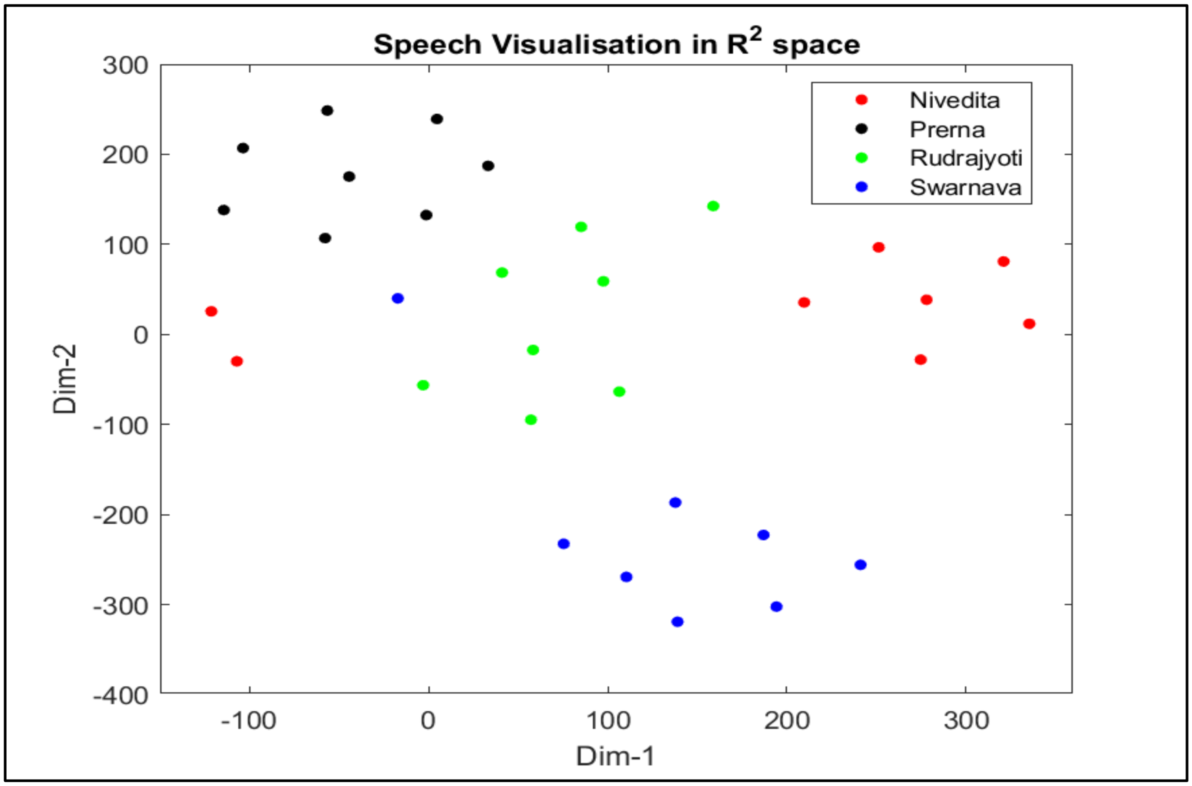
	Nivedita-Afternoon	Nivedita-Morning	Prerna-Afternoon	Prerna-Morning	Rudrajyoti-Afternoon	Rudrajyoti-Morning	Swarnava-Afternoon	Swarnava-Morning
Nivedita-Afternoon	0	0.8293	0.7272	0.6039	0.8357	0.6947	0.8777	0.7758
Nivedita-Morning	0.8293	0	1.158	1.069	1.284	1.142	0.8913	1.012
Prerna-Afternoon	0.7272	1.158	0	0.1374	0.8503	0.7261	1.148	1.194
Prerna-Morning	0.6039	1.069	0.1374	0	0.8144	0.6624	1.09	1.114
Rudrajyoti-Afternoon	0.8357	1.284	0.8503	0.8144	0	0.3831	1.13	1.157
Rudrajyoti-Morning	0.6947	1.142	0.7261	0.6624	0.3831	0	0.9966	1.012
Swarnava-Afternoon	0.8777	0.8913	1.148	1.09	1.13	0.9966	0	0.3471
Swarnava-Morning	0.7758	1.012	1.194	1.114	1.157	1.012	0.3471	0

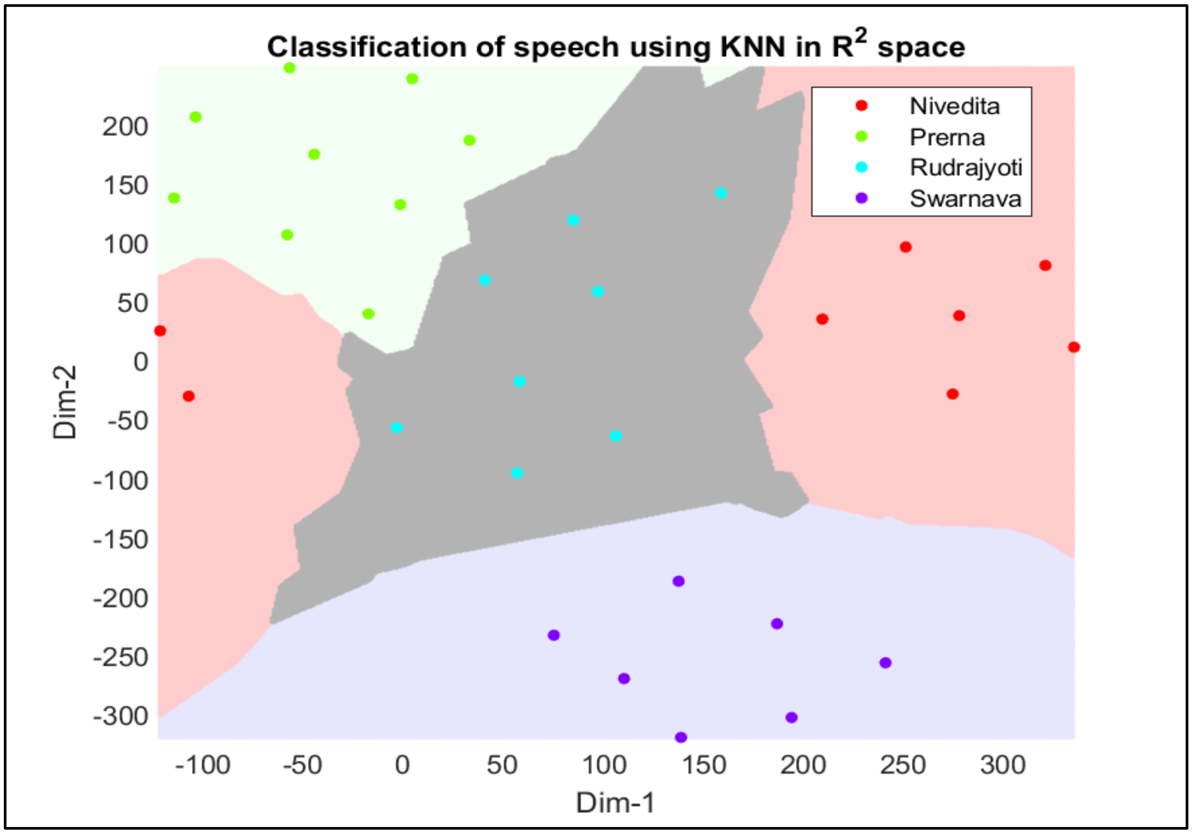
KNN - Uniform Filters





KNN - Non Uniform Filters





DISCUSSIONS: -

Euclidean Distance Based Similarity Heatmap: - We obtain heat-maps using Euclidean Distance as a parameter. Two types of Heatmaps were plotted, one using Uniform Filter Placement and another using Non-uniform Filter Placement. For each, we plot two plots; one using all samples and another using the average of all morning samples for each user and average of all afternoon samples for each user. In the heatmap, light colour corresponds to less Euclidean distance and greater correlation. Darker colour means less correlation. The following observations were made: -

- The correlation between the voice of the two female speakers (Nivedita and Prerna) is high. Similarly, the correlation between the voice of two male speakers (Rudrajyoti and Swarnava) is also high. But the inter-gender voice correlation is low, as expected.
- The correlation between the morning samples for each person is high. Similarly, the correlation between afternoon samples was also high. But the correlations between the morning and afternoon samples were relatively lower. Although, it is not very low as they are still the voice of the same individual.
- Finally the correlation between multiple utterances was found to be very high (low Euclidean distance) for the same person but low for different persons.

We do not observe any noticeable differences with uniform and non-uniform filter placement or the number of filters in the heatmap plot, however, during classification, we can observe a significant difference in performance (Explained in detail later).

Speech Verification: We obtain the following performance from linear and non-linear placement of filter in filter-banks comprising of 20 filters. We observe that one sample is misclassified in the training examples after training with uniformly distributed filters, while for the non-uniform filter-bank, we get all samples correctly classified, thus giving 100% accuracy. However, we get 100% validation accuracies in both cases.

Filter Placement (20 filters)	**Training Accuracy**	**Validation Accuracy**
Non-uniform filter placement	100%	100%
Uniform filter placement	96.43%	100%

Do note that the decision boundary plot in the diagrams is done separately using data reduced to 2-dimensional feature space, and hence can be inaccurate as compared to the original classification done using the entire 20-dimensional feature vector.

Experimentation with Filter Bank Length and Filter Width

Filter Bank containing 10 filters -

Filter Placement (10 filters)	**Training Accuracy**	**Validation Accuracy**
Non-uniform filter placement	96.43%	75%
Uniform filter placement	85.71%	75%

Filter bank contains 50 filters -

Filter Placement (50 filters)	**Training Accuracy**	**Validation Accuracy**
Non-uniform filter placement	89.29%	75%
Uniform filter placement	100%	100%

When the number of filters is kept constant at 20, it is observed that we obtain better identification accuracy when more filters are used in 0-1000Hz range, compared to the uniform filter placement. The fundamental frequency of human speech lies in the 100Hz-300Hz range, hence most of the speech energy is concentrated below 1000 Hz, Therefore, increasing the filter resolution in this region results in the **variation in spectral energy getting captured more effectively**.

Furthermore, it is observed that with decrease in number of filters (20 to 10), the identification accuracy decreases. This is reasonable, as with decrease in the number of features per speech samples, it is harder to categorize different samples. But, with increase in the number of filter banks (20 to 50), the accuracy decreases unexpectedly.

The reason may be attributed to the very low passband width requirement of the Butterworth filters below 1000Hz. For non-uniform pole placement and number of filters being 50, the passband width of each filter is 40 Hz. Thus, the realized 5th order Butterworth filter has significant passband attenuation as evident from the `freqz` plot below. Thus, the spectral energy is not captured properly and hence accuracy decreases.

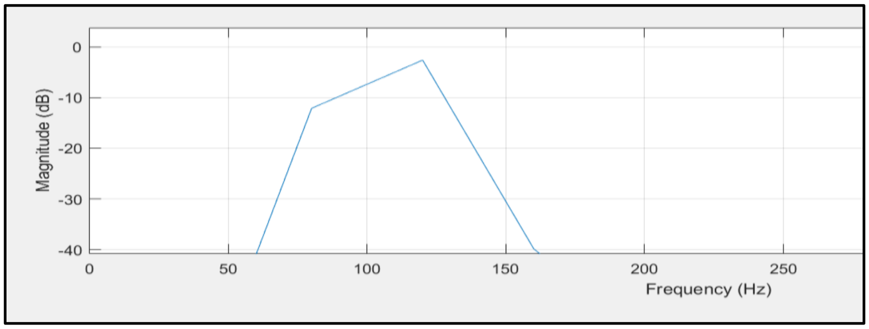


Figure 1 Frequency Response 3rd Filter Bank when 50 filters are placed non-uniformly

False acceptance and False rejection rates achieved:

We vary the hyper-parameter for threshold scaling factor to modify the threshold value for classification that affects its performance. The following tables show the deviation present in the training (train + validation) voices and testing voices for the non-uniform filter-bank comprising of 20 filters-

Train Voices	Nivedita	Prerna	Rudrajyoti	Swarnava
Deviation(max)	1.47	1.35	1.62	1.18

Test Voices	Aishik	Debjoy	Nivedita	Rudrajyoti	Swarnava
Deviation	1.89	1.78	0.59	1.00	1.16

The above two tables show the deviation from the mean of all voices for a particular speaker. For the Train Voices table, the deviation is the maximum obtained of all the 8 speech samples. For the Test Voices, the deviation is taken from the mean of the voice belonging to the speaker classified by the KNN-classifier. The red highlighted columns belong to the people not in the registered speaker list. We see that selecting the **threshold scaling factor (TH_{OPT}) as 1.7**, we can achieve **zero** false acceptance and false rejection rates, since this can -

(a) Allow all training samples as true, since the maximum deviation (1.62) lies within threshold (1.7).
(b) Reject all false test samples, since both deviation values (1.89 and 1.78) are greater than threshold.
(c) Allow all true test samples, since both deviation values (0.59, 1.00 and 1.16) are less than threshold.

However, for different filter-bank settings, non-zero false acceptance and rejection rates might be obtained, but we can expect them to be negligibly low.

Performance of the Pitch Detection Algorithm:

This whole algorithm is implemented in MATLAB using the `pitch()` function with a certain level of abstraction. The window length was chosen to be 3% of the sampling frequency F_S while the overlap length was chosen as 2.5% of F_S. This is to ensure that the window contains the optimal number of samples to uniquely obtain the fundamental frequency of speech irrespective of the sampling frequency. These were provided as input parameters to the function. After detecting the pitch, the harmonic ratio is determined using `harmonicRatio()` function which returns the **ratio of the fundamental frequency's power to the total power in an audio frame**. This ratio is usually very low during the unvoiced part of the speech signal because of the presence of low amplitude noise. Thus, the detected pitch is considered invalid where the harmonic ratio is below a certain threshold (0.5).

The obtained plot contained multiple spikes. This is because the higher octaves might be wrongly detected as the fundamental frequency in some regions where the amplitude corresponding to true fundamental frequency is low. Thus, **we have used a median filter (of length 10 samples), to filter the regional spikes**. The estimated pitch is calculated as the mean of the time vs pitch sequence.

Observation :- It is reasonably observed that the pitch of the girls are in the interval 200 Hz – 250 Hz while the pitch of the boys are in the interval 100 Hz – 150 Hz. This proves that girls voices are generally more shrill compared to the boys.

It is also generally observed that for the same person, the voice samples recorded in the morning have slightly lower pitch and the variation in pitch is more, while the voice samples recorded in the afternoon have slightly higher pitch and the pitch is more stable. This might be caused due to the accumulation of laryngeal mucus over the vocal tract in the morning due to lack of vocal activity while sleeping.

Limitations of HPS algorithm for pitch detection :- Some nice features of this method include: it is computationally inexpensive, reasonably resistant to additive and multiplicative noise, and adjustable to different kinds of inputs. For instance, we could change the number of compressed spectra to use, and we could replace the spectral multiplication with a spectral addition. However, since human pitch perception is basically logarithmic, this means that low pitches may be tracked less accurately than high pitches.

Alternative pitch detection algorithm :- Fundamentally, this algorithm exploits the fact that a periodic signal, even if it is not a pure sine wave, will be similar from one period to the next. This is true even if the amplitude of the signal is changing in time, provided those changes do not occur too quickly. To detect the pitch, we take a window of the signal, with a length at least twice as long as the longest period that we might detect. Using this section of signal, we generate the autocorrelation function $r(s)$ defined as the sum of the pointwise absolute difference between the two signals over some interval. Intuitively, it should make sense that as the shift value s begins to reach the fundamental period of the signal T, the difference between the shifted signal and the original signal will begin to decrease. We can detect this value by differentiating the autocorrelation function and then looking for a change of sign, which yields critical points. We then look at the direction of the sign change across points (positive difference to negative), to take only the minima. We then search for the first minimum below some threshold, i.e., the minimum corresponding to the smallest s. The location of this minimum gives us the fundamental period of the windowed portion of signal, from which we can easily determine the frequency.

------------------------------ End of Report ------------------------------