Speaker recognition using LPC and MFCC features with KNN classifier

Vineet Puri
Department of Electrical and Computer Engineering
Western University
London, Ontario
Email: vpuri8@uwo.ca

***Abstract*— With increasing adaptation of the voice based virtual assistants and other voice-based systems, speaker recognition has become an active area of research in recent years. Different machine learning models can be used to carry out speaker recognition. But before applying any machine learning model, we first need to identify and extract features from speech signals. In this project, different feature extraction algorithms have been implemented on a dataset of speech samples of different people. The classification has been carried out using k-NN and finally the results for different feature extraction combinations are compared. Effect of noise on accuracies of models has also been explored.**

Keywords*— *Feature extraction, MFCC, LPC, reflection coefficients, k-NN, speaker recognition

I. INTRODUCTION

While speech recognition involves identifying the words being spoken, speaker recognition involves identifying the speaker using different techniques. Some systems are text based in which the speaker needs to speak a predefined set of words, while others are non-text based in which speaker can speak anything. Both speech recognition and speaker recognition have advanced a lot in recent years, and this has mostly been possible because of machine learning methodologies. The success of machine learning models depends upon the data being fed into it. A speech signal in raw form as an input can be very difficult to train on because it leads to high data volume. It is much better if the information in signal can be represented with lesser number of elements. This is achieved through feature extraction. There are many popular techniques for features extraction for speech signals. Linear Predictive Coding (LPC) and Mel Frequency Cepstral Coefficients (MFCC) are the ones which have been used in this project.

Rest of the report is divided into four parts. In next part, related background has been discussed for this problem and how others have approached this topic. In third part, different algorithms used in the project have been discussed with implementation details. In fourth part, simulation results have been presented and analyzed. Fifth part includes the references. An appendix containing all the simulation results has been provided at the end.

II. BACKGROUND

Speaker recognition has applications in a wide range of fields. Smart home solutions and virtual assistants like Alexa work on speech-based commands and it can be helpful if such assistants can identify different members of the house and then customize results according to them. Speaker recognition can also be used for identity verification. Such systems can verify the identity of a person based on speech. Another application can be automatic transcription of a meeting with speaker identification. Such a system can tell us who spoke what in a meeting.

Through feature extraction, the speech signal is converted into a parametric representation, which can be further analyzed and classified [1]. Many studies involving different feature extraction algorithms have been carried out. It has been found that each algorithm has its own merits and demerits. LPC has been found to be useful for high frequency speech signals [2] but it struggles with words having same vowel sounds [3]. MFCC has been found to be very efficient and accurate with low complexity, but its results are affected by background noise. We have used both these techniques in this project and compared their performance. The performance and accuracy of speech recognition systems can vary greatly due to background noises and linear distortions [2]. The problem of noise can be addressed at the feature extraction stage of the system by selecting an algorithm that produces features that are resistant to background noise changes but is still able to capture the salient speech information [4]. In this project, the effect of noise on feature extraction algorithms has also been checked and discussed.

III. SYSTEM DESCRIPTION AND ALGORITHMS

In this project we have used the Census Database (also known as AN4 Database) from the CMU Robust Speech Recognition Group [5]. Initially speech samples for 10 people from the database were selected and samples for eleventh person have been recorded in my own voice. In later part of the project, samples from 10 more people have been included to see how it affects the accuracies. Dataset was divided into train and test set. Different feature extraction algorithms and their combinations were used. Extracted features from the training set were then used to train the k-NN model. Same feature extraction was applied on the test set and then trained model was used to carry out the prediction. The overall system can be described in the form of a diagram as shown in fig.1.

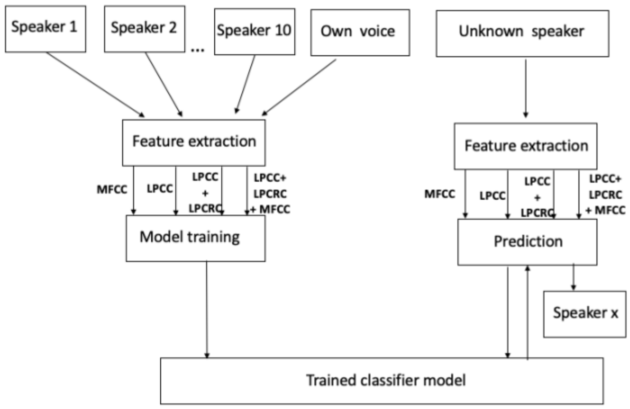


Fig. 1. System description

Different steps followed in the process are described below-
(a) The dataset was downloaded with different folders for 10 different speakers. The folder names acted as labels.
(b) 10 samples were recorded in my own voice and added to the dataset as eleventh speaker.
(c) The dataset was split into train and test set.
(d) Training files were loaded one after another for training.
(e) Each file was split into frames with overlapping windows.
(f) Required feature extraction was carried out per frame.
(g) Each frame was allotted the same label as the original audio file to which they belonged.
(h) Extracted features were collected frame wise in a matrix for all the audio files.
(i) Normalization of features was carried out by subtracting the mean and dividing by the standard deviation.
(j) The features were then used to train the k-NN classifier model with the number of neighbors set to five.
(k) Five-fold cross validation was performed, and validation accuracy was calculated.
(l) The model was used to make predictions on the test set. The predictions were made per frame and frame wise test accuracy was calculated.
(m) Prediction for an entire audio sample file was obtained from its frames by taking mode of the frame wise predictions. Sample file wise accuracies were calculated.

Different algorithms used in the project are explained below-

(a) **LPCC and LPC Reflection Coefficients**- Linear predictive coding represents the spectral envelope of a speech signal in a compressed form, using the information of a linear predictive model [6]. It can be carried out in a forward adaptive manner and in a backward adaptive manner. The process followed in the project to get linear predictive coding coefficients and linear predictive coding reflection coefficients is shown in fig.2. First the signal was split into frames, then a standard pre-emphasis filter was used to provide gain to the higher frequencies. Then a hamming window was applied to deal with the sidelobes. After that autocorrelation was performed and finally the coefficients were calculated using the Levinson-Durbin algorithm. The LPC order was taken as sampling rate in kHz + 3. In our case, samples had a sampling rate of 16 kHz, so order for LPC was taken as 19.

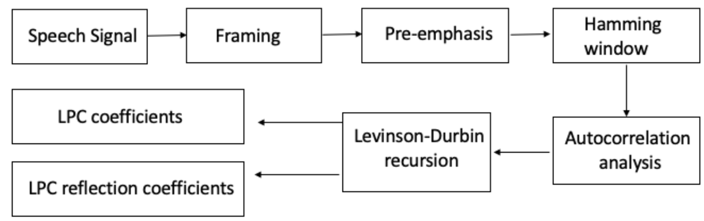


Fig. 2. Linear predictive coding process

(b) **MFCC**- Mel-frequency Cepstrum is a representation of the short-term power spectrum of a sound [7]. It is based on a linear cosine transform of a log power spectrum on a nonlinear Mel scale of frequency. It is designed to mimic the human auditory system. The procedure followed in the project to extract Mel-frequency Cepstrum Coefficients is shown in fig.3. First, the pre-emphasis was done to provide gain to the higher frequencies. Next, the signal was split into frames, then Hann window was applied to deal with the sidelobes. After this short time Fourier transform was carried out and absolute values were found. Then the signal frame was passed through a Mel filter bank. The last two stages involving log and DCT were carried out using a cepstralCoefficients() function in the MATLAB.

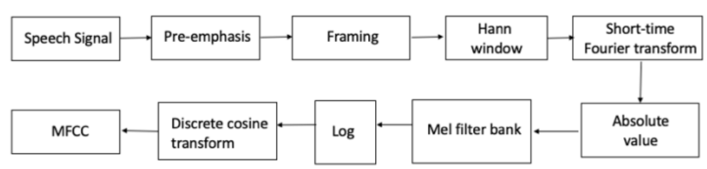


Fig. 3. MFCC process

(c) **k-NN**- KNN is a supervised machine learning algorithm which can be used for both classification as well as regression. In this project, it has been used for classification. In classification, 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 [8]. The value of k has been chosen to be 5 in this project. This value is a hyper parameter and can be tuned to get better results. An example of k-NN is shown in fig.4. Here if k = 3 then the green circle is assigned to the red triangles but if k = 5 then it is assigned to the blue squares.

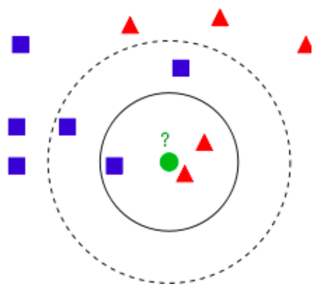


Fig. 4. k-NN example [8]

IV. SIMULATION RESULTS

(i) **LPC testing**- First, the LPC function was tested on a sample file to confirm that it can estimate the signal with sufficient accuracy. The results are shown in fig. 5.

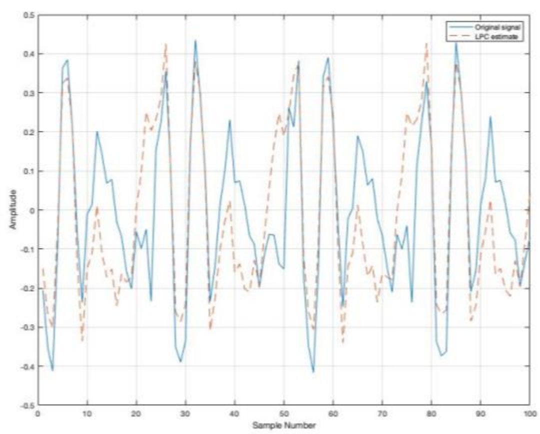


Fig. 5. LPC estimate

(ii) **Feature extraction with 11 speakers**- The model was trained with different feature extraction combinations for 11 speakers. Confusion matrices were plotted for validation as well as testing process for frame wise predictions.

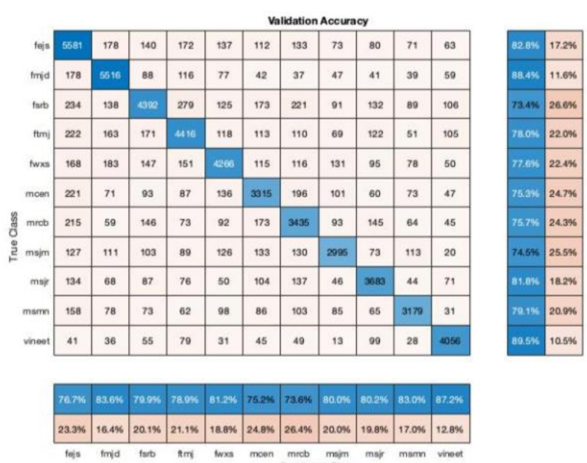


Fig. 6. Frame wise validation accuracies for LPCC

The confusion matrix for LPCC validation is shown in fig. 6. It can be seen that the validation accuracies range from 73% to 89% for the 11 classes. The last class called "vineet" is for my own voice samples.

Frame wise test accuracies for LPC is shown in fig. 7. Test accuracies are lower as compared to validation accuracies. Similar plots were generated for all the feature extraction methods and their combinations. Full results are provided in the appendix of the report.

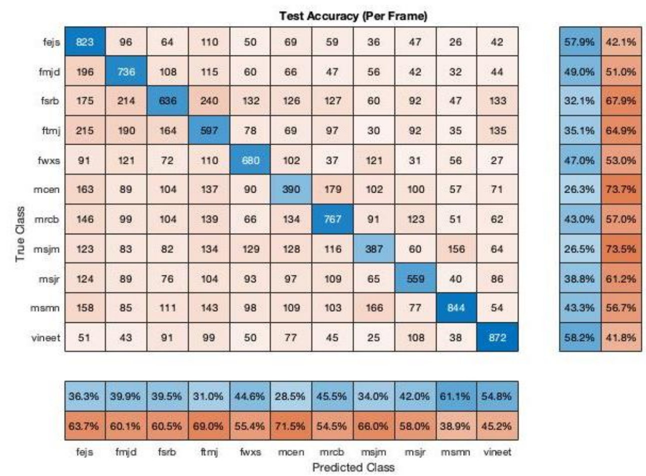


Fig. 7. Frame wise test accuracies for LPCC

Frame wise validation accuracies and test accuracies for all the feature combinations are compared in table 1. When a combination of LPCC and LPC reflection coefficients was used as a feature set then it led to increase in accuracy as compared to the case when only LPCC was used. Using MFCC alone resulted in higher accuracies as compared to the LPCC and LPCRC combination. Finally, when all the features were used together then the validation accuracy improved even further although test accuracy was almost the same as in MFCC alone. The audio file wise test accuracies were 100% for all the cases and this result was further explored after adding noise in the next stage.

Table 1

Features used (11 classes)	Validation accuracy for frames	Test accuracy for frames
LPCC	79.88%	41.26%
LPCC+LPCRC	87.56%	47.09%
MFCC	87.96%	57.99%
MFCC+LPC+LPCRC	92.21%	57.76%

(iii) Feature extraction with 11 speakers and added noise-

The best two models from the previous stage were chosen for this phase. AWGN noise was added to the speech signals to get 10 dB SNR and 5 dB SNR signals. Feature extraction was then carried out on the noisy signals and predictions were made. Results are shown in table 2.

There was a huge drop in accuracies for both the models as the amount of noise was increased. When all the three features were used together for training then the validation accuracies were higher, but test accuracies were lower as compared to the case when MFCC was used alone.

On the other hand, test accuracies for whole sample files were no longer 100% when 5 dB SNR was used as shown in fig.8 and fig.9. The performance was impacted more when using MFCC alone.

Table 2

Noise level	Features used (11 classes)	Validation accuracy for frames	Test accuracy for frames
10 dB SNR	MFCC	66.88%	44.10%
	MFCC+LPCC+ LPCRC	74.61%	34.40%
5 dB SNR	MFCC	62.67%	40.75%
	MFCC+LPCC+ LPCRC	72.23%	31.53%

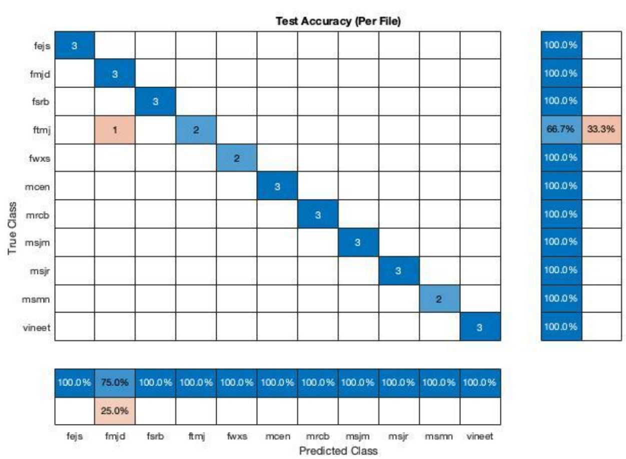


Fig. 8. Sample wise test accuracies with 5dB SNR for all features used

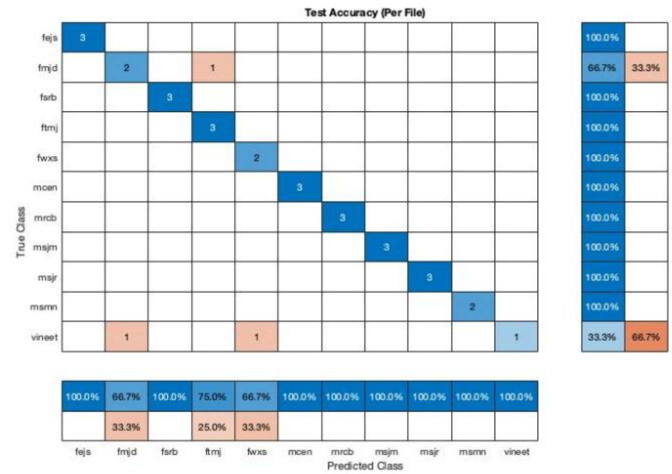


Fig. 9. Sample wise test accuracies with 5dB SNRfor MFCC only

(iv) **Feature extraction with 21 speakers-** In the last phase, the number of speakers or classes were increased from 11 to 21. Again, one class had my own voice samples. Two best models were used for feature extraction. No noise was added to the signals. Results are shown in table 3. The test accuracy was almost the same for both the models. Audio file wise accuracies were again 100% and it may be because no noise was added in this phase. Full results are presented in the appendix.

Table 3

Features used (21 classes)	Validation accuracy for frames	Test accuracy for frames
MFCC	83.02%	41.42%
MFCC+LPC+LPCRC	89.42%	41.66%

V. Conclusion

In this project, linear prediction and MFCC were used to extract features from speech signals. These features were then used to train a machine learning model to predict the speaker. Different combinations of features were compared for accuracies. The impact of noise was also studied on some models, and it was found that noise had a serious impact on the performance of all the models. Overall, the best model was found to be the one when all the three features i.e., LPC coefficients, LPC reflection coefficients and MFCC were used together. This model gave better accuracy on sample file classification with 5 dB SNR as compared to the case when using MFCC alone. Although, frame wise accuracies were better for MFCC with 5 dB SNR. Finally, the number of classes were also increased, and the models were compared. Again, the model with all the features included gave better validation accuracy.

In future, hyper parameter tuning can be performed to improve the accuracies. For example, different k values can be tried for the k-NN model. Other machine learning models can also be tried with same features. Additional feature extraction algorithms like RASTA and ZCR can also be implemented and compared with the results obtained in this project.

VI References

[1] Shumaila Iqbal, Tahira Mahboob, Malik Sikandar and Hayat Khiyal, "Voice Recognition using HMM with MFCC for Secure ATM", International Journal of Computer Science Issues (IJCSI), vol. 8, no. 6, pp. 297-303, November 2011.

[2] K. Gupta and D. Gupta, "An analysis on LPC, RASTA and MFCC techniques in Automatic Speech recognition system," 2016 6th International Conference - Cloud System and Big Data Engineering (Confluence), 2016, pp. 493-497, doi:10.1109/CONFLUENCE.2016.7508170

[3] Vibha Tiwari, "MFCC and Its Applications in Speaker Recognition", International Journal on Emerging Technologies, vol. 1, no. 1, pp. 19-22, 2010.

[4] U. Shrawankar and V. Thakare, "Feature Extraction for a Speech Recognition System in Noisy Environment: A Study," 2010 Second International Conference on Computer Engineering and Applications, 2010, pp. 358-361, doi: 10.1109/ICCEA.2010.76.

[5] "CMU Sphinx Group - Audio Databases." Accessed November 21, 2021. http://www.speech.cs.cmu.edu/databases/an4/.

[6] “Linear_predictive_coding” Accessed November 25, 2021, Online : https://en.wikipedia.org/wiki/Linear_predictive_coding

[7] “Mel-frequency cepstrum” Accessed November 30, 2021, Online : https://en.wikipedia.org/wiki/Mel-frequency_cepstrum

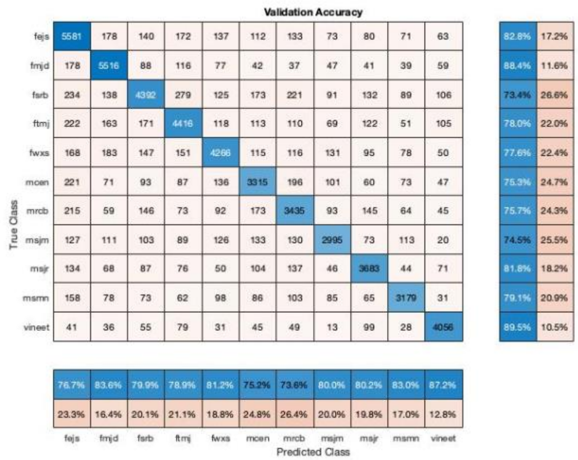
[8] “k-nearest neighbors algorithm” Accessed November 30, 2021, Online : https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

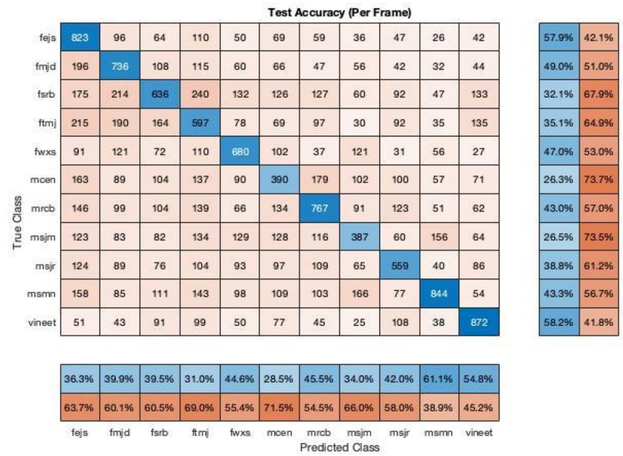
Appendix
Confusion matrices for different models

(a) Experiments with 11 speakers and no noise

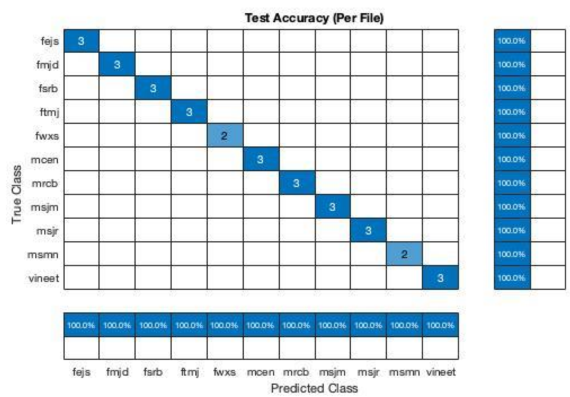
(i) LPCC only

Per frame accuracies-



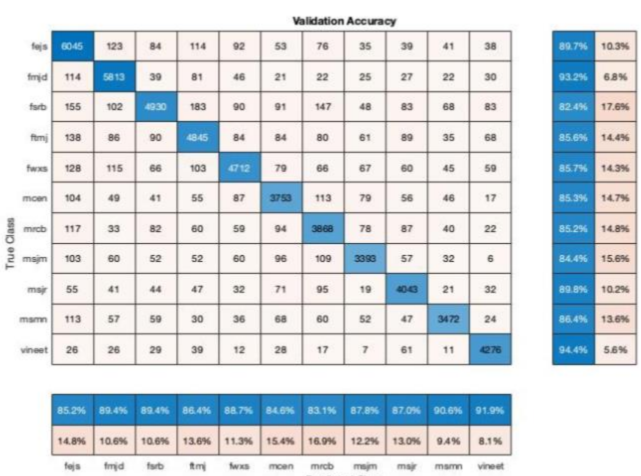


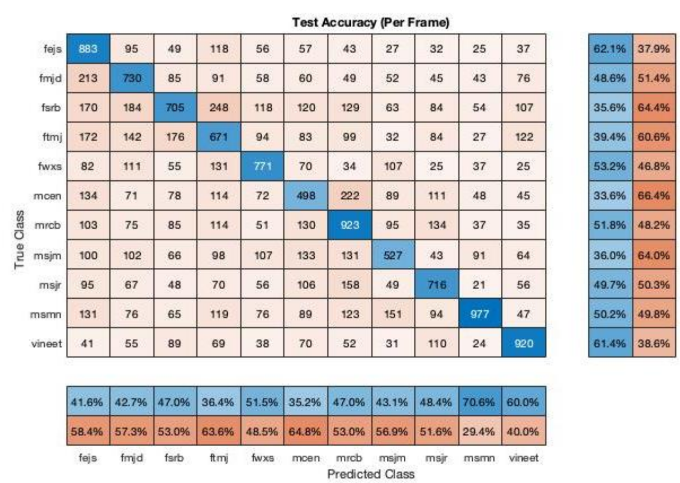
Per audio file accuracy-



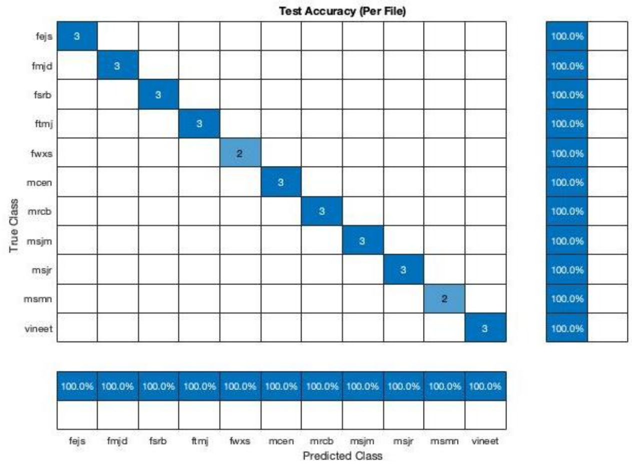
(ii) LPCC + LPC reflection coefficients

Per frame accuracies-

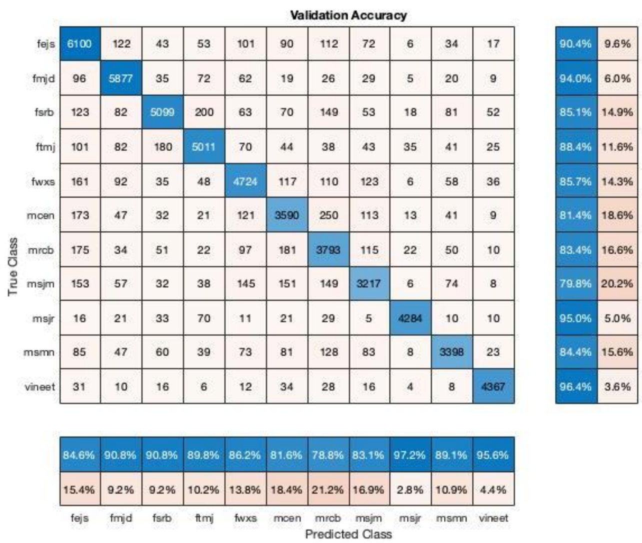


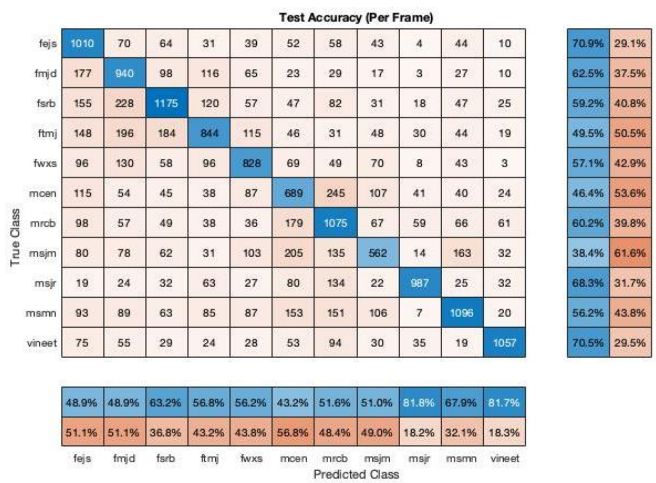


Per audio file accuracy-

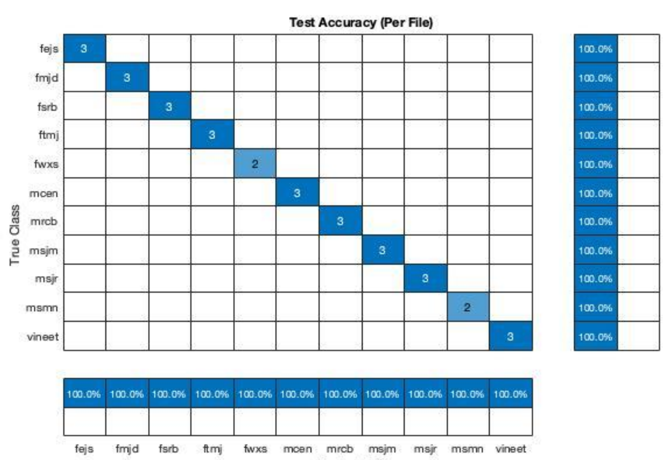


(iii) MFCC only
Per frame accuracies-



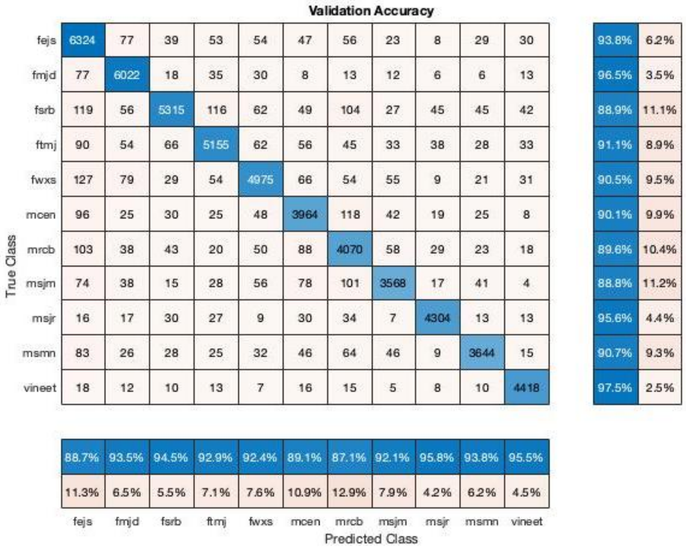


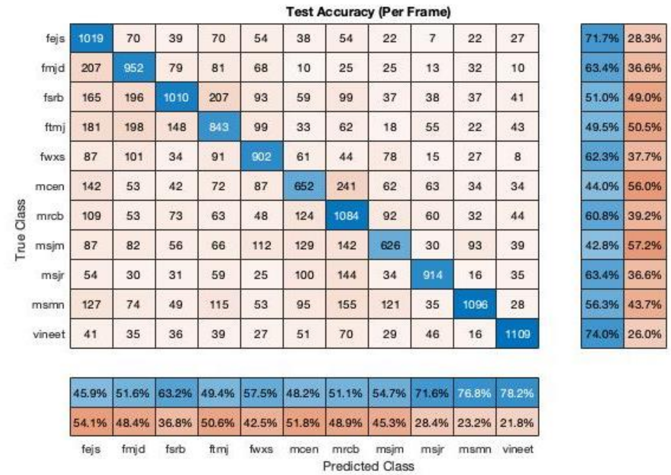
Per audio file accuracy-



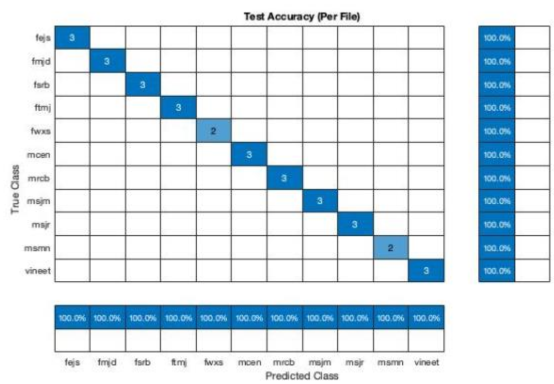
(iv) LPCC + LPC reflection coefficients + MFCC

Per frame accuracies-





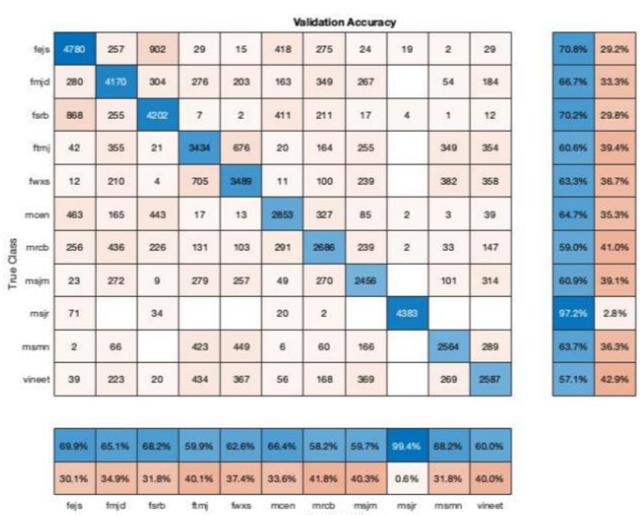
Per audio file accuracy-

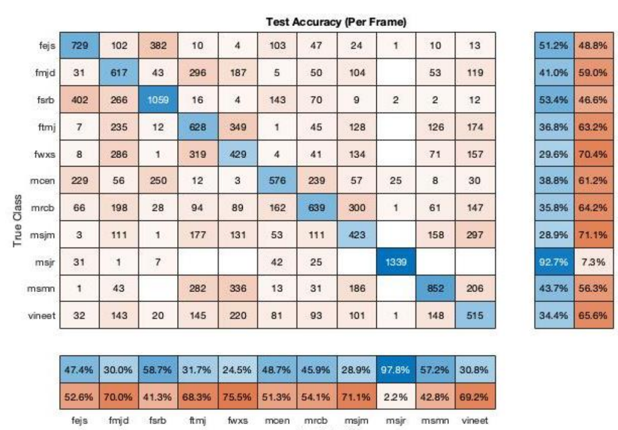


(b) Experiments with 11 speakers and 10 dB SNR

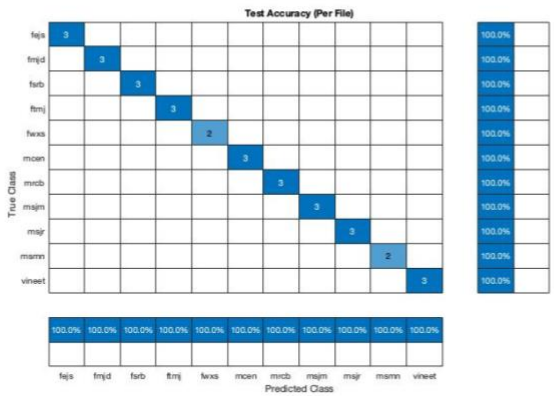
(i) MFCC only

Per frame accuracies-



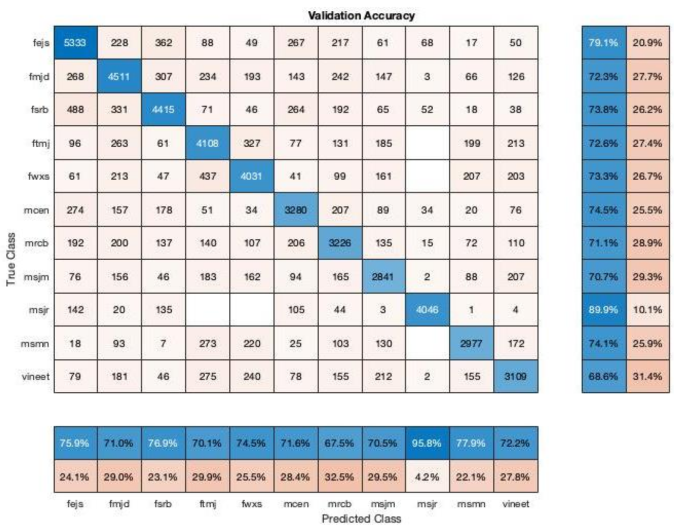


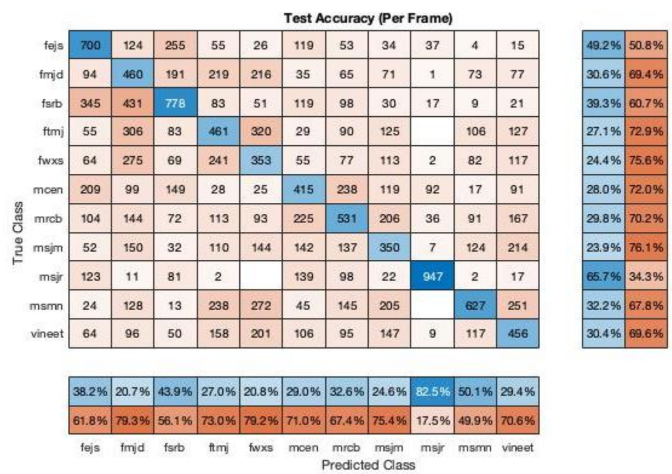
Per audio file accuracy-



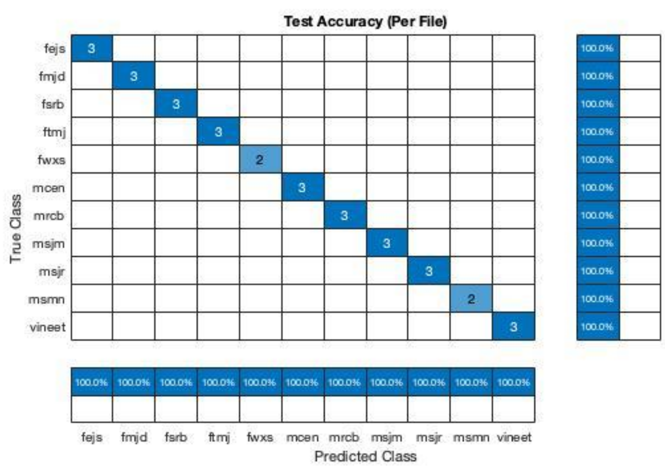
(ii) LPCC + LPC reflection coefficients + MFCC

Per frame accuracies-





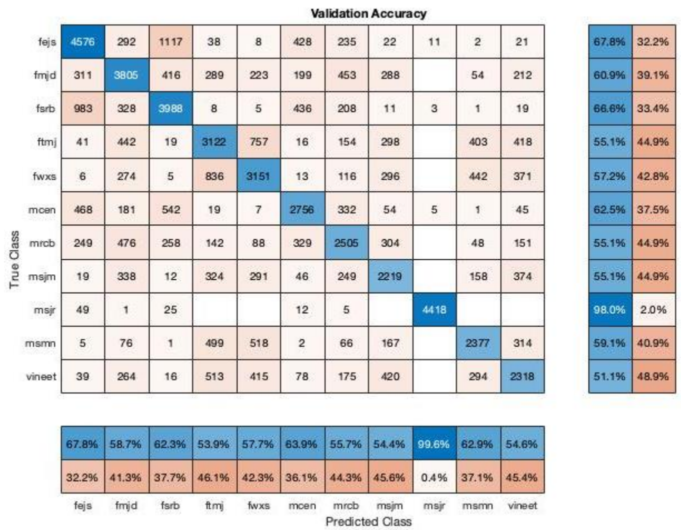
Per audio file accuracy-

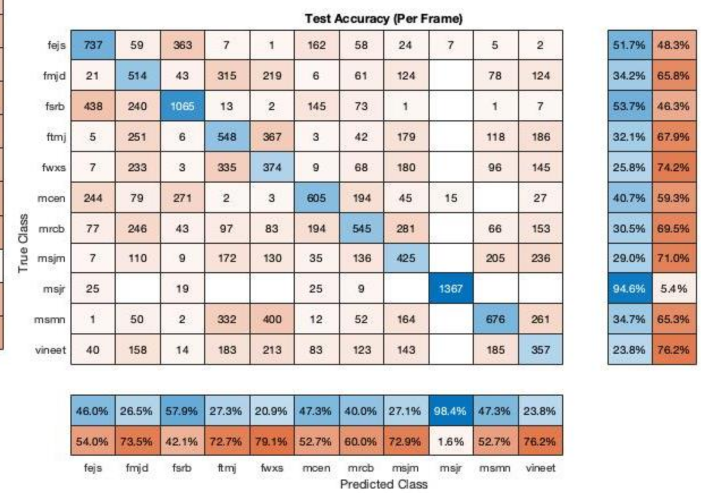


(c) Experiments with 11 speakers and 5 dB SNR

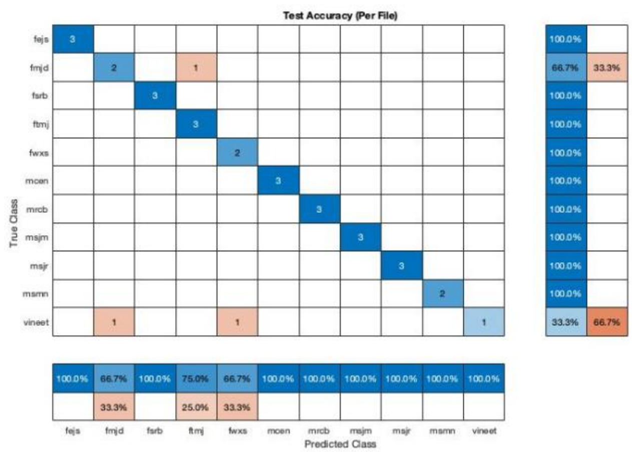
(i) MFCC only

Per frame accuracies-



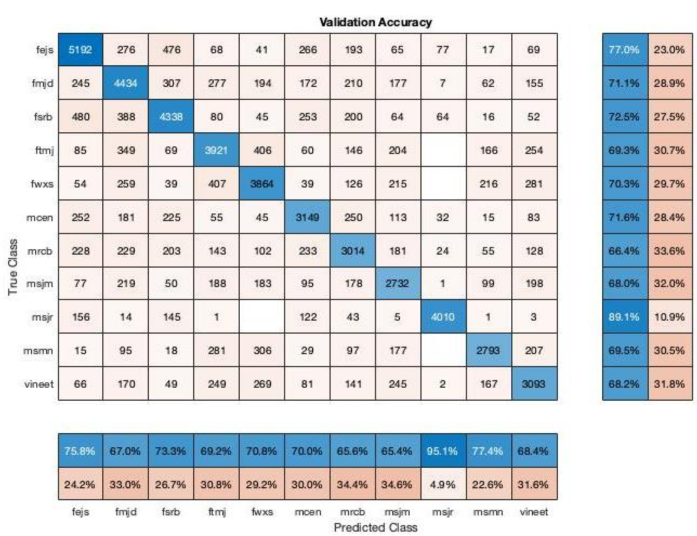


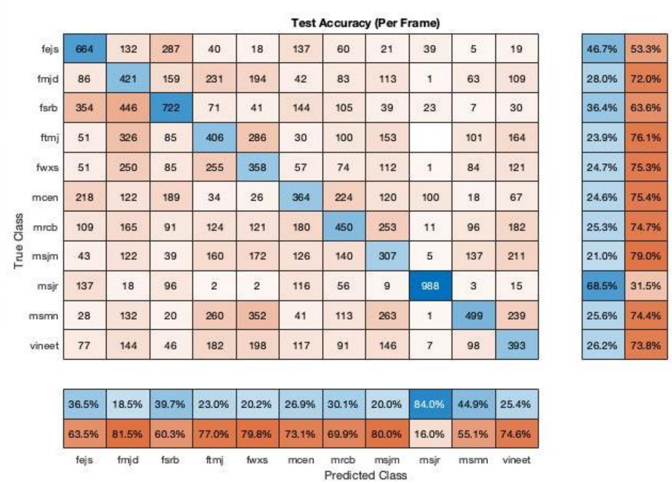
Per audio file accuracy-



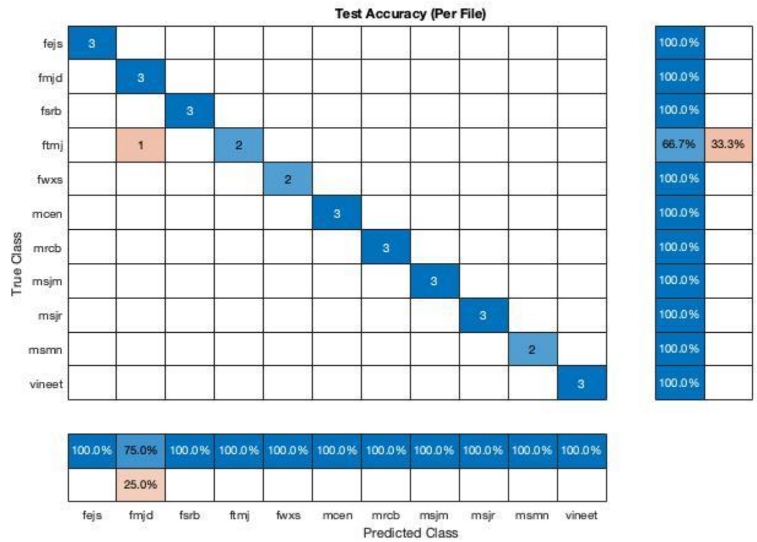
(ii) LPCC + LPC reflection coefficients + MFCC

Per frame accuracies-

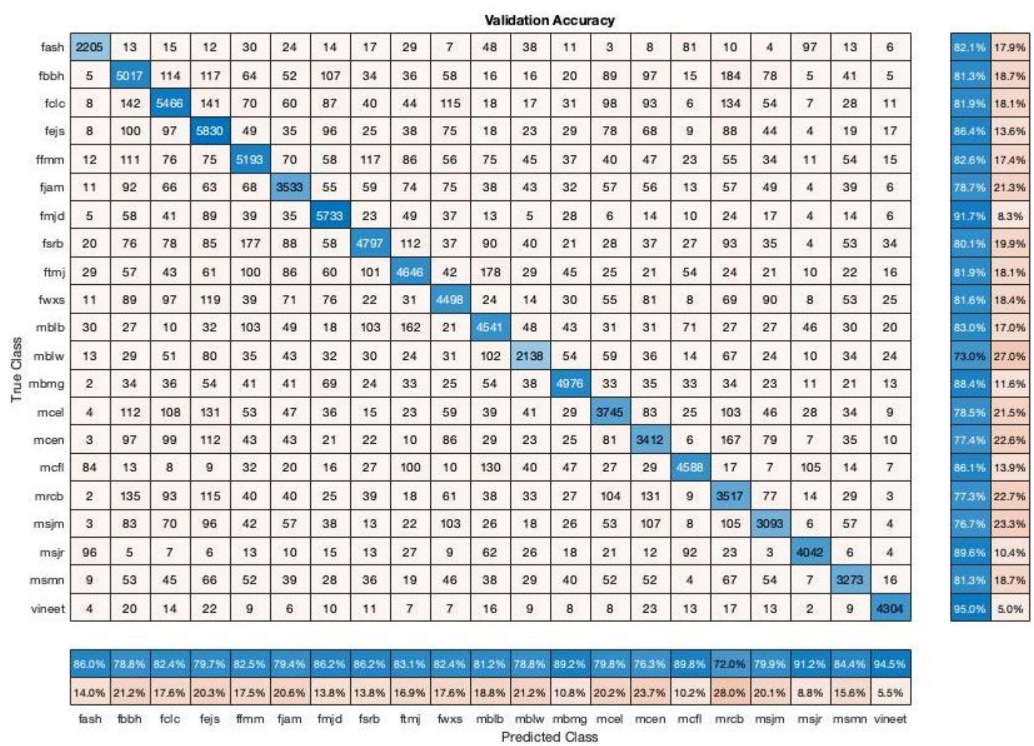


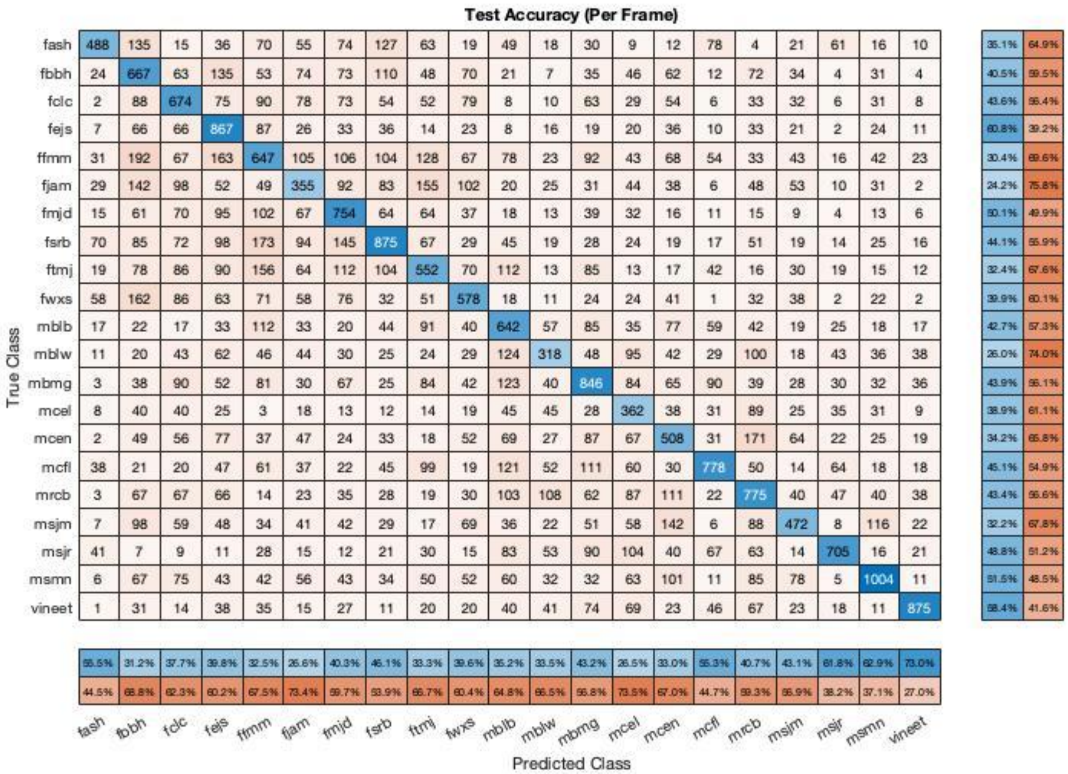


Per audio file accuracy-

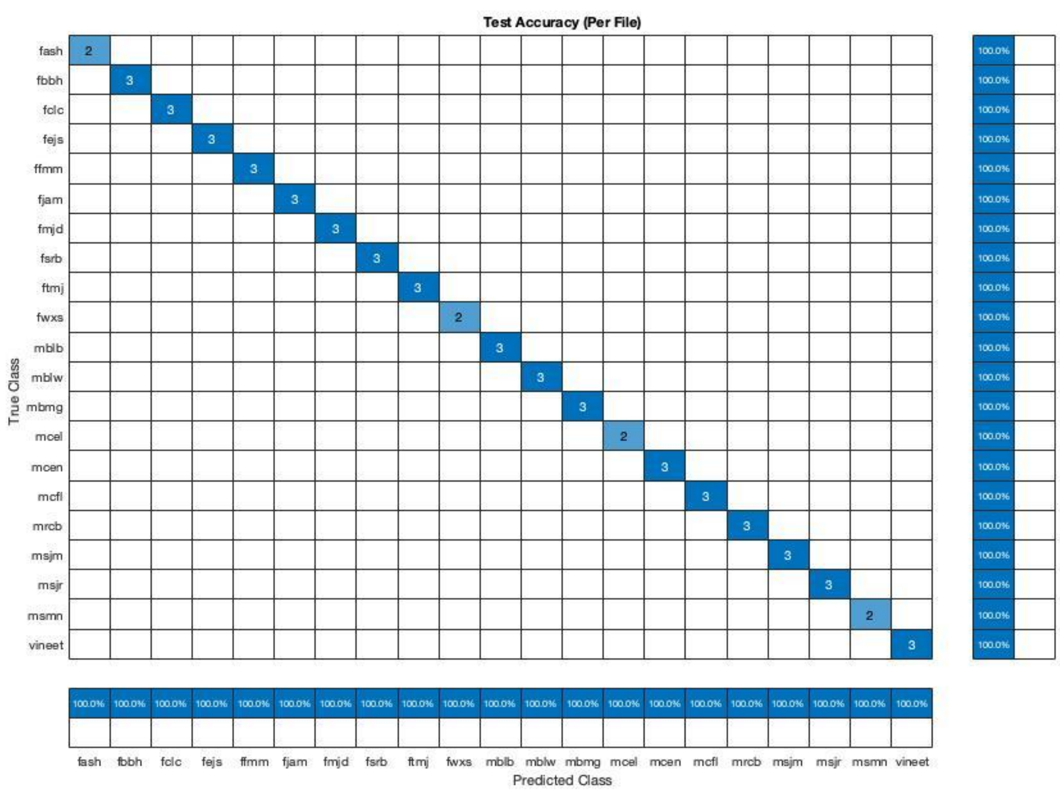


(d) Experiments with 21 speakers and no noise
(i) MFCC only
Per frame accuracies-

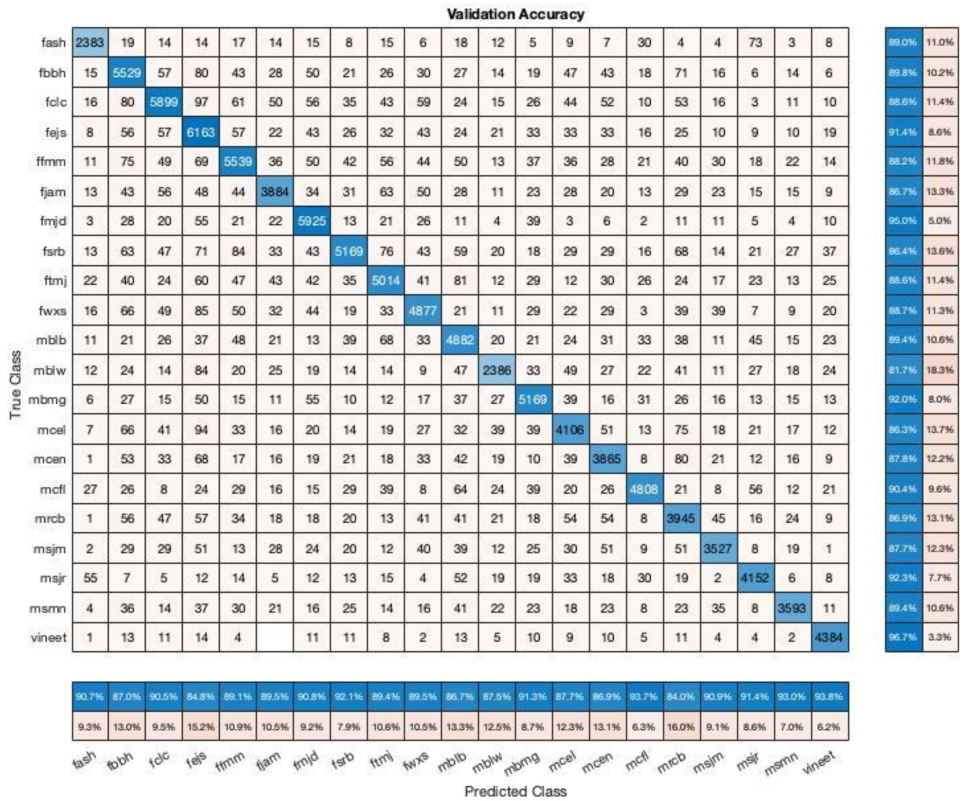


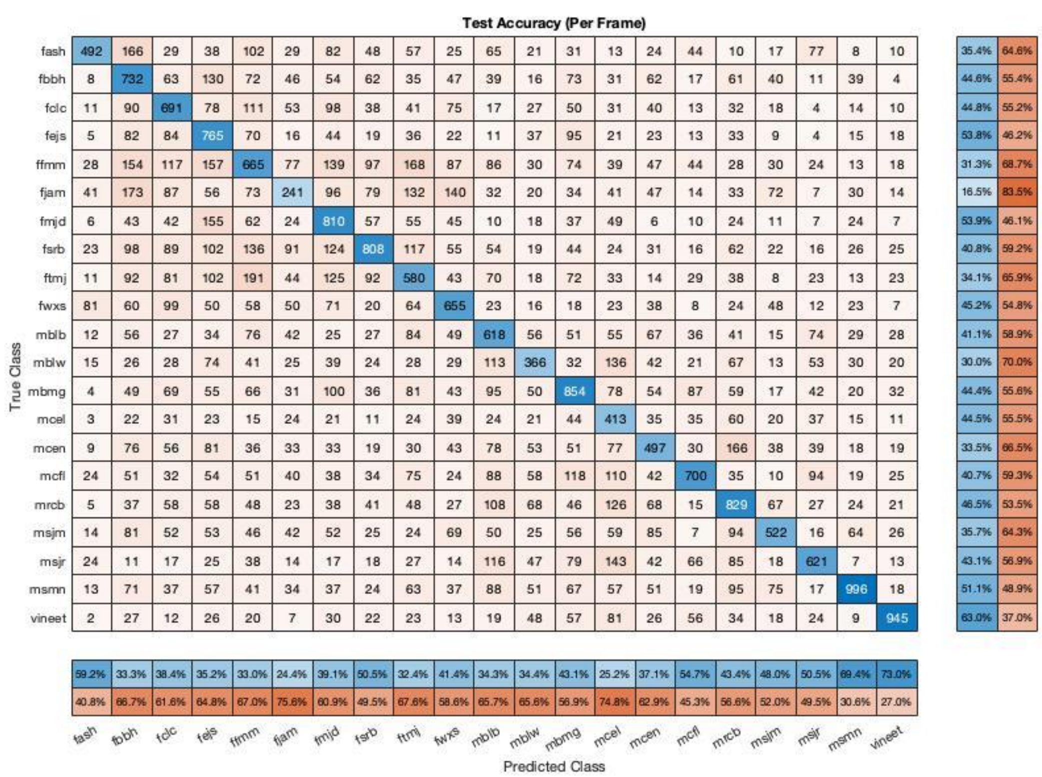


Per audio file accuracy-



(ii) LPCC + LPC reflection coefficients + MFCC
Per frame accuracies-





Per audio file accuracy-