

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.

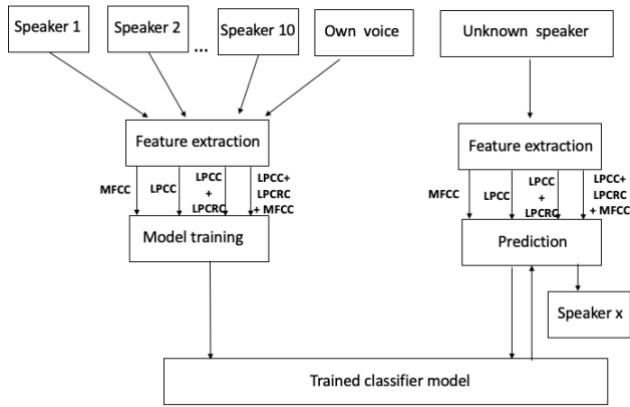


Fig. 1. System description

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.

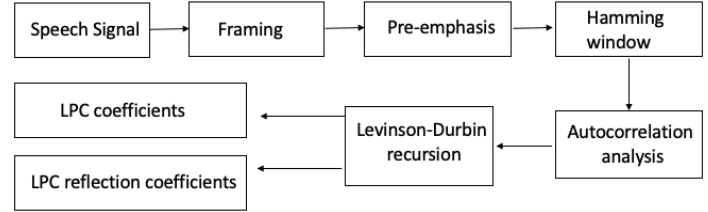


Fig. 2. Linear predictive coding process

(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.

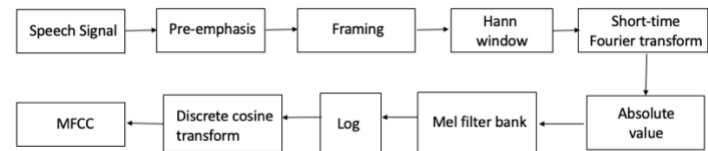


Fig. 3. MFCC process

(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.

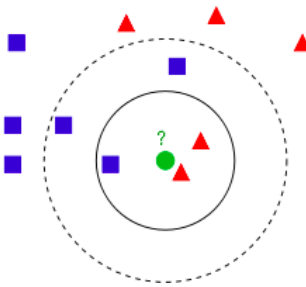


Fig. 4. k-NN example [8]

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.

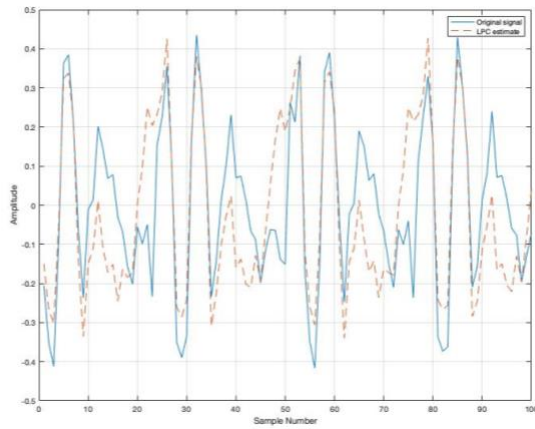


Fig. 5. LPC estimate

(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.

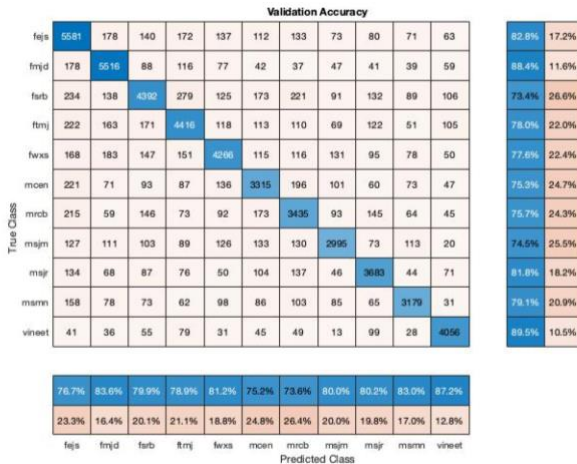


Fig. 6. Frame wise validation accuracies for LPCC

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.

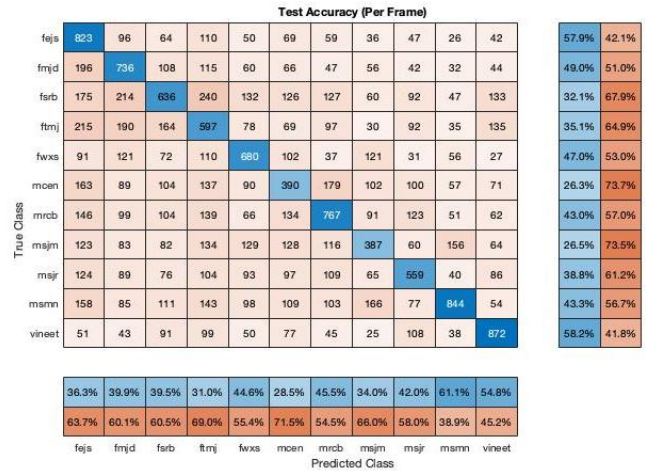


Fig. 7. Frame wise test accuracies for LPCC

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%

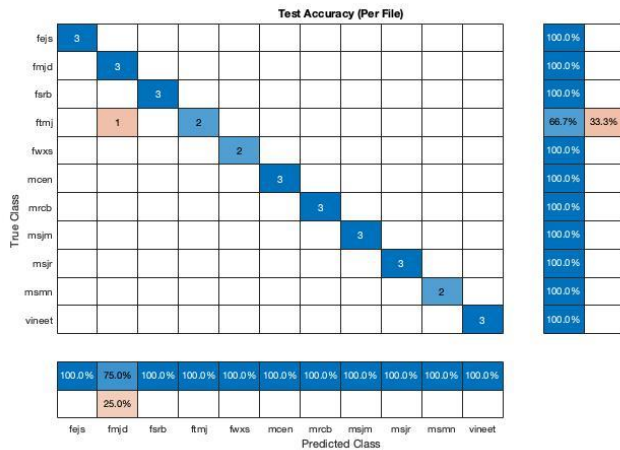


Fig. 8. Sample wise test accuracies with 5dB SNR for all features used

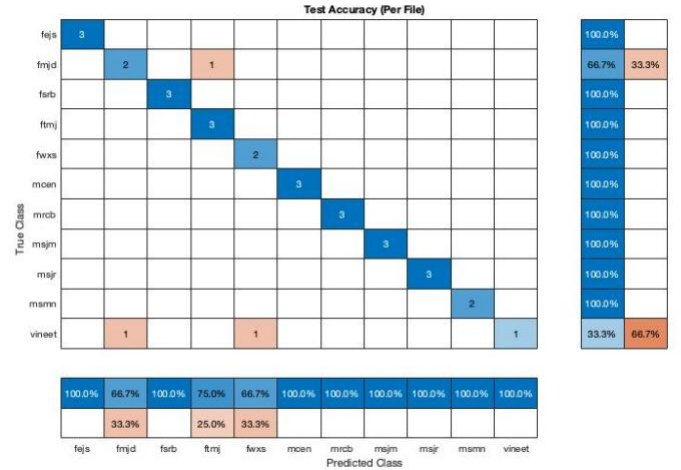


Fig. 9. Sample wise test accuracies with 5dB SNR for MFCC only

(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-

Validation Accuracy													
fejls	5561	178	140	172	137	112	133	73	80	71	63	82.8%	17.2%
fmjld	178	5515	88	116	77	42	37	47	41	39	59	88.4%	11.6%
farfb	234	138	4390	279	125	173	221	91	132	89	106	73.4%	26.6%
ftmj	222	163	171	4416	118	113	110	69	122	51	105	78.0%	22.0%
fwxs	168	183	147	151	4598	115	116	131	95	78	50	77.8%	22.4%
moen	221	71	93	87	136	3315	196	101	60	73	47	75.3%	24.7%
mrcb	215	59	146	73	92	173	3435	93	145	64	45	75.7%	24.3%
msjm	127	111	103	89	126	133	130	2995	73	113	20	74.5%	25.5%
msjr	134	68	87	76	50	104	137	46	3683	44	71	81.8%	18.2%
msmn	158	78	73	62	98	86	103	85	65	3179	31	79.1%	20.9%
vineet	41	36	55	79	31	45	49	13	99	28	4056	89.5%	10.5%
	fejls	fmjld	farfb	ftmj	fwxs	moen	mrcb	msjm	msjr	msmn	vineet		
	76.7%	83.6%	79.9%	78.9%	81.2%	75.2%	73.6%	80.0%	80.2%	83.0%	87.2%		
	23.3%	16.4%	20.1%	21.1%	18.8%	24.8%	26.4%	20.0%	19.8%	17.0%	12.8%		

Test Accuracy (Per Frame)													
fejls	823	96	64	110	50	69	59	36	47	26	42	57.9%	42.1%
fmjld	196	736	108	115	60	66	47	56	42	32	44	49.0%	51.0%
farfb	175	214	636	240	132	126	127	60	92	47	133	32.1%	67.9%
ftmj	215	190	164	597	78	69	97	30	92	35	135	35.1%	64.9%
fwxs	91	121	72	110	680	102	37	121	31	56	27	47.0%	53.0%
moen	163	89	104	137	90	390	179	102	100	57	71	26.3%	73.7%
mrcb	146	99	104	139	66	134	767	91	123	51	62	43.0%	57.0%
msjm	123	83	82	134	129	128	116	387	60	156	64	26.5%	73.5%
msjr	124	89	76	104	93	97	109	65	559	40	86	38.8%	61.2%
msmn	158	85	111	143	98	109	103	166	77	844	54	43.3%	56.7%
vineet	51	43	91	99	50	77	45	25	108	38	672	58.2%	41.8%
	fejls	fmjld	farfb	ftmj	fwxs	moen	mrcb	msjm	msjr	msmn	vineet		
	36.3%	39.9%	39.5%	31.0%	44.6%	28.5%	45.5%	34.0%	42.0%	61.1%	54.8%		
	63.7%	60.1%	60.5%	69.0%	55.4%	71.5%	54.5%	66.0%	58.0%	38.9%	45.2%		

Per audio file accuracy-

Test Accuracy (Per File)													
fejls	3											100.0%	
fmjld		3										100.0%	
farfb			3									100.0%	
ftmj				3								100.0%	
fwxs					2							100.0%	
moen						3						100.0%	
mrcb							3					100.0%	
msjm								3				100.0%	
msjr									3			100.0%	
msmn										2		100.0%	
vineet											3	100.0%	
	fejls	fmjld	farfb	ftmj	fwxs	moen	mrcb	msjm	msjr	msmn	vineet		
	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%		

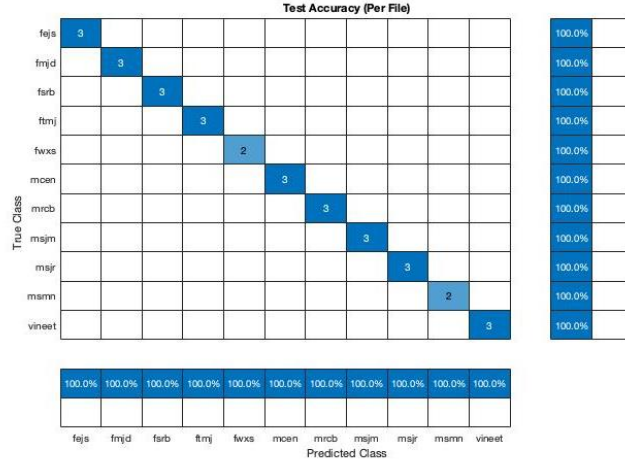
(ii) LPCC + LPC reflection coefficients

Per frame accuracies-

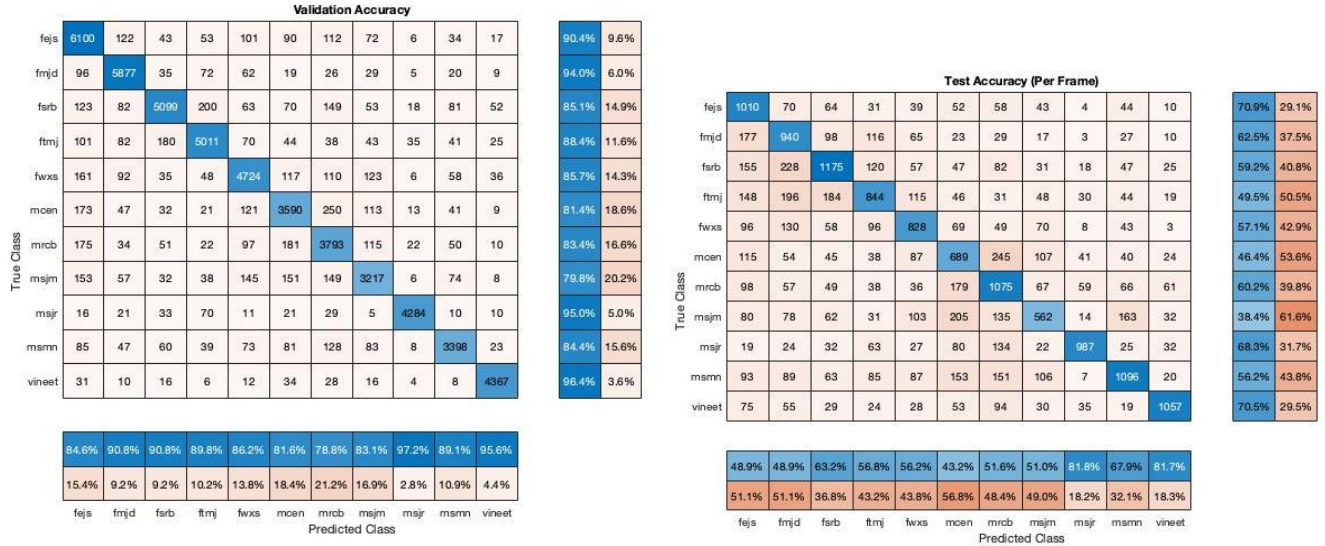
Validation Accuracy													
fejls	6045	123	84	114	92	53	76	35	39	41	38	89.7%	10.3%
fmjld	114	5813	39	81	46	21	22	25	27	22	30	93.2%	6.8%
farfb	155	102	4930	183	90	91	147	48	83	68	83	82.4%	17.6%
ftmj	138	86	90	4945	84	84	80	61	89	35	68	85.6%	14.4%
fwxs	128	115	66	103	4712	79	66	67	60	45	59	85.7%	14.3%
moen	104	49	41	55	87	3753	113	79	56	46	17	85.3%	14.7%
mrcb	117	33	82	60	59	94	3868	78	87	40	22	85.2%	14.8%
msjm	103	60	52	52	60	96	109	3393	57	32	6	84.4%	15.6%
msjr	55	41	44	47	32	71	95	19	4043	21	32	89.8%	10.2%
msmn	113	57	59	30	36	68	60	52	47	3472	24	86.4%	13.6%
vineet	26	26	29	39	12	28	17	7	61	11	4276	94.4%	5.6%
	fejls	fmjld	farfb	ftmj	fwxs	moen	mrcb	msjm	msjr	msmn	vineet		
	85.2%	89.4%	89.4%	86.4%	88.7%	84.6%	83.1%	87.8%	87.0%	90.6%	91.9%		
	14.8%	10.6%	10.6%	13.6%	11.3%	15.4%	16.9%	12.2%	13.0%	9.4%	8.1%		

Test Accuracy (Per Frame)													
fejls	883	95	49	118	56	57	43	27	32	25	37	62.1%	37.9%
fmjld	213	730	85	91	58	60	49	52	45	43	76	48.6%	51.4%
farfb	170	184	705	248	118	120	129	63	84	54	107	35.6%	64.4%
ftmj	172	142	176	671	94	83	99	32	84	27	122	39.4%	60.6%
fwxs	82	111	55	131	771	70	34	107	25	37	25	53.2%	46.8%
moen	134	71	78	114	72	498	222	89	111	48	45	33.6%	66.4%
mrcb	103	75	85	114	51	130	923	95	134	37	35	51.8%	48.2%
msjm	100	102	66	98	107	133	131	527	43	91	64	36.0%	64.0%
msjr	95	67	48	70	56	106	158	49	716	21	56	49.7%	50.3%
msmn	131	76	65	119	76	89	123	151	94	977	47	50.2%	49.8%
vineet	41	55	89	69	38	70	52	31	110	24	920	61.4%	38.6%
	fejls	fmjld	farfb	ftmj	fwxs	moen	mrcb	msjm	msjr	msmn	vineet		
	41.6%	42.7%	47.0%	36.4%	51.5%	35.2%	47.0%	43.1%	48.4%	70.6%	60.0%		
	58.4%	57.3%	53.0%	63.6%	48.5%	64.8%	53.0%	56.9%	51.6%	29.4%	40.0%		

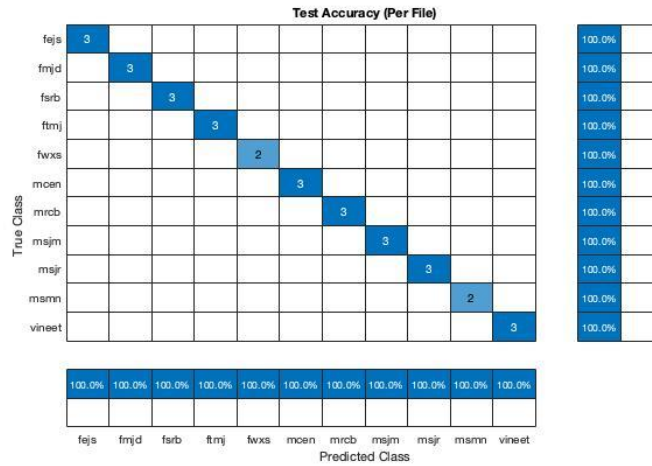
Per audio file accuracy-



(iii) MFCC only
Per frame accuracies-

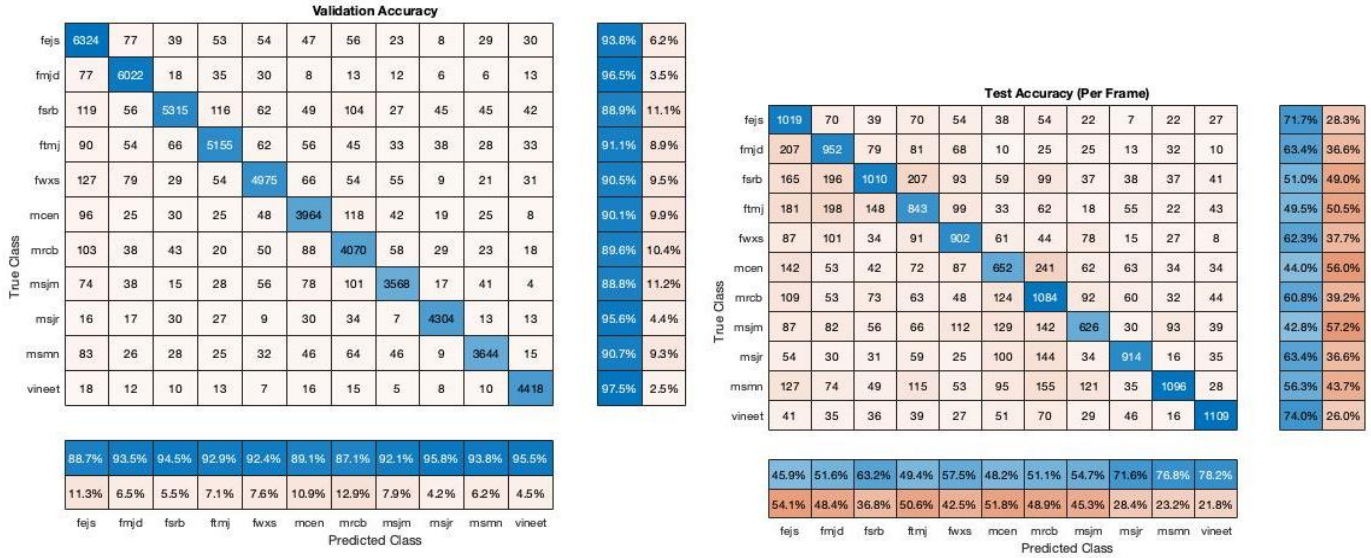


Per audio file accuracy-

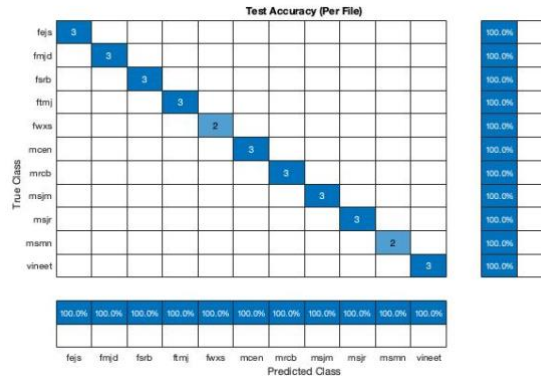


(iv) LPCC + LPC reflection coefficients + MFCC

Per frame accuracies-



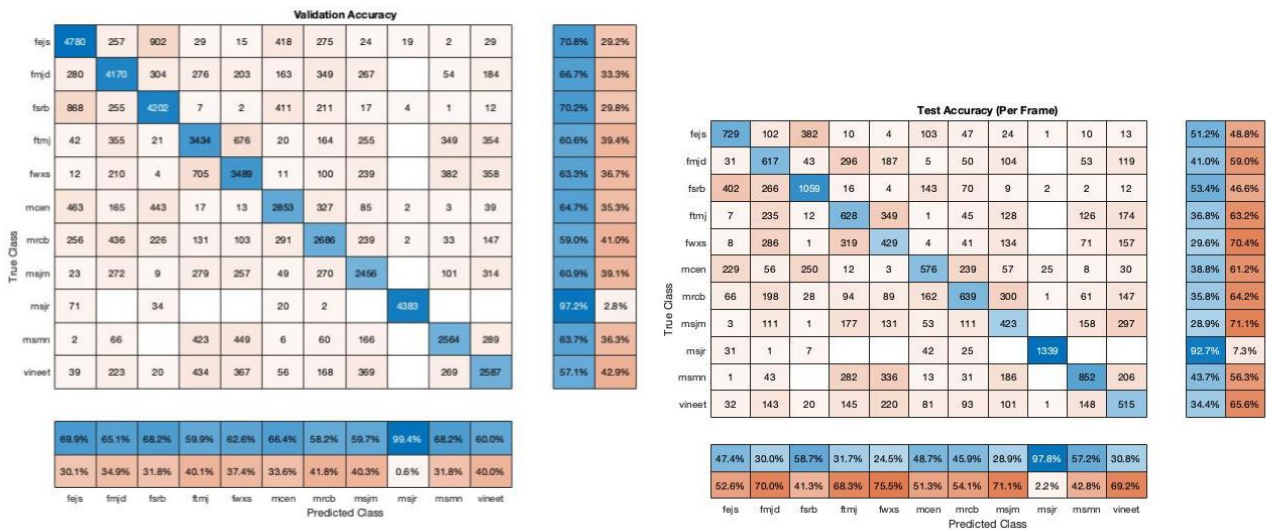
Per audio file accuracy-



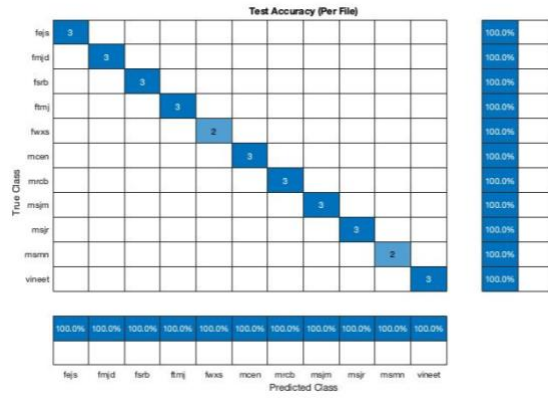
(b) Experiments with 11 speakers and 10 dB SNR

(i) MFCC only

Per frame accuracies-

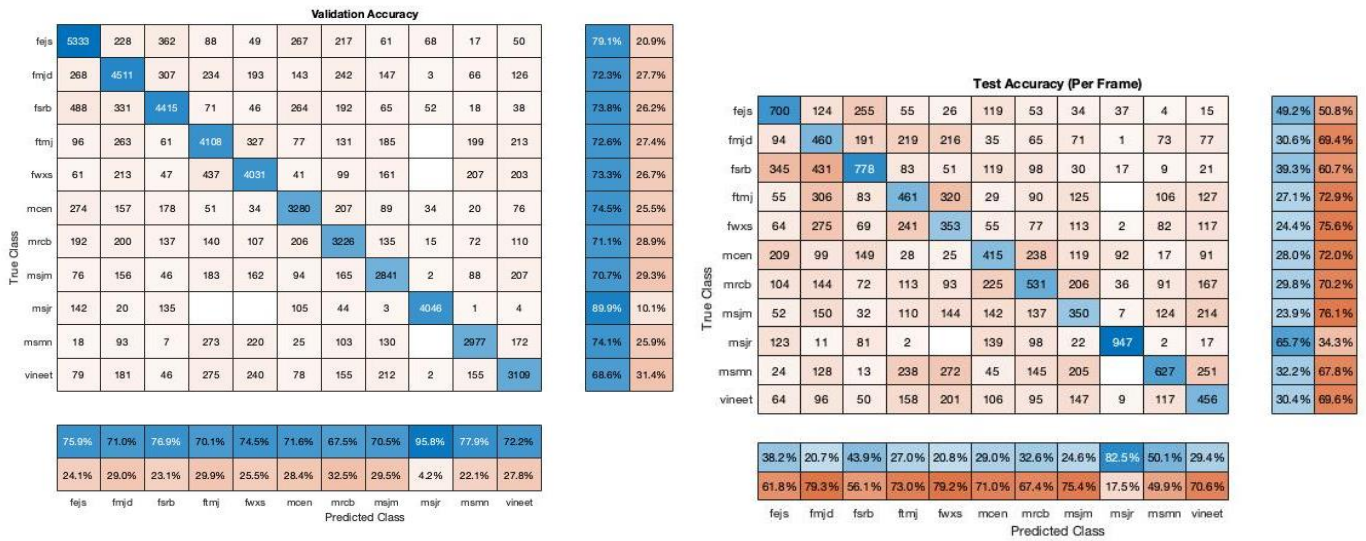


Per audio file accuracy-

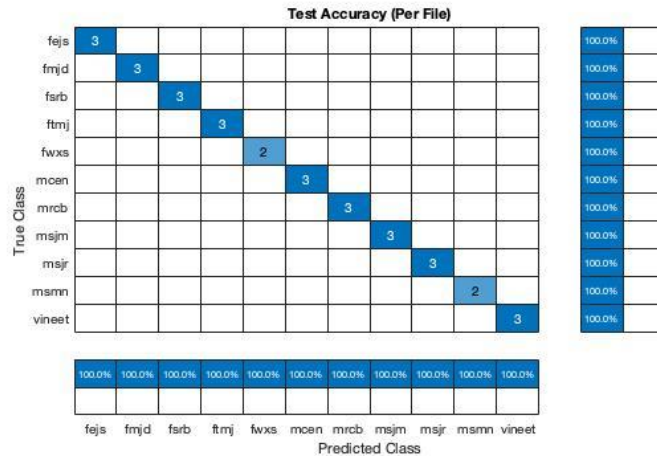


(ii) LPCC + LPC reflection coefficients + MFCC

Per frame accuracies-



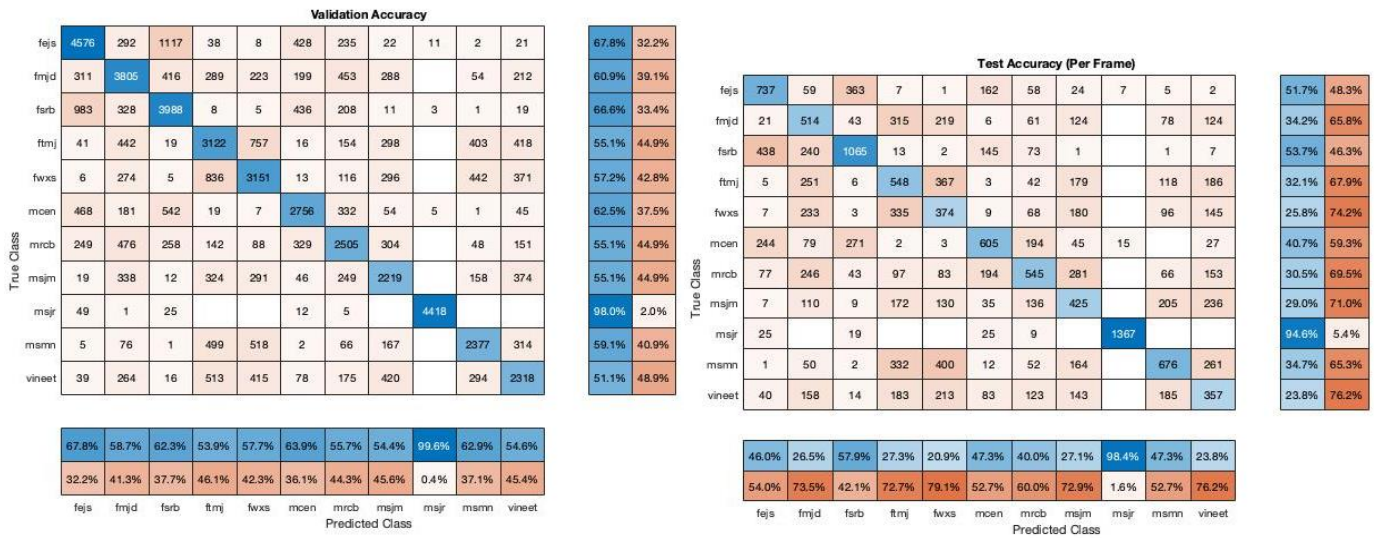
Per audio file accuracy-



(c) Experiments with 11 speakers and 5 dB SNR

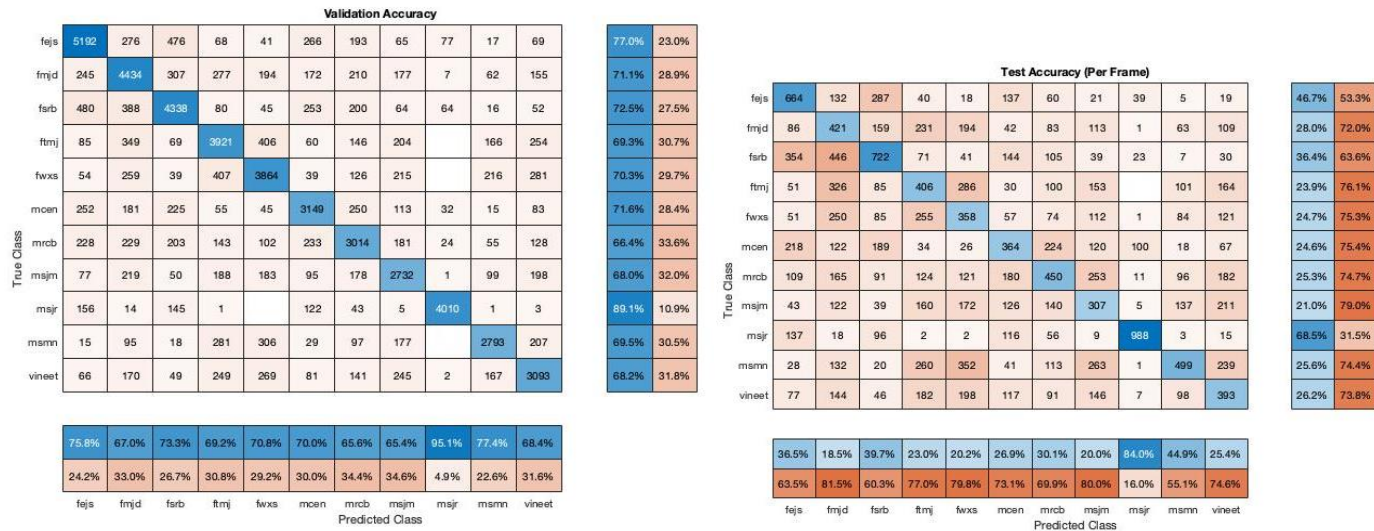
(i) MFCC only

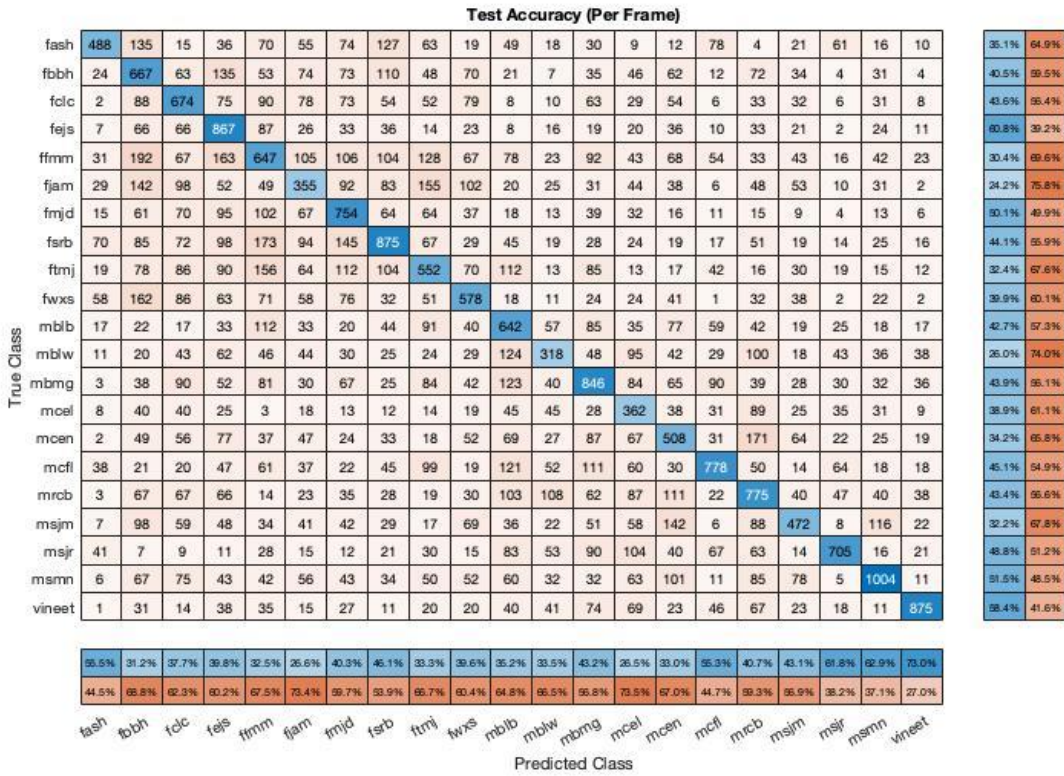
Per frame accuracies-



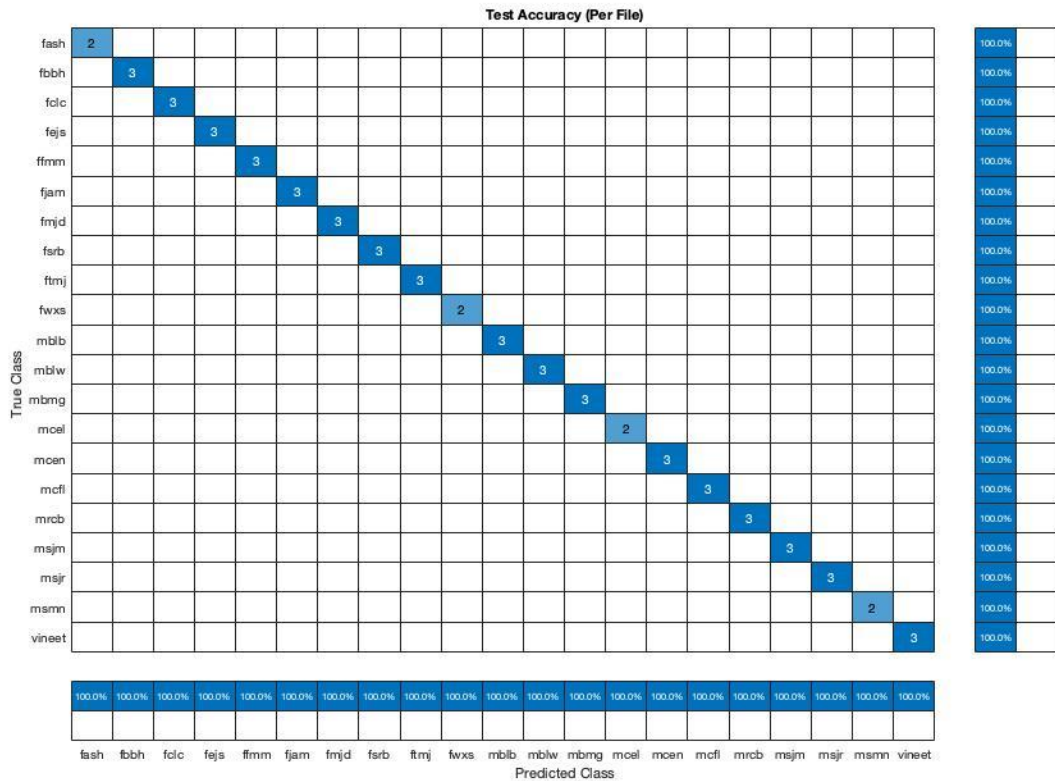
(ii) LPCC + LPC reflection coefficients + MFCC

Per frame accuracies-





Per audio file accuracy-



Per frame accuracies-

		Validation Accuracy																							
True Class	fash	2383	19	14	14	17	14	15	8	15	6	18	12	5	9	7	30	4	4	73	3	8	89.0%	11.0%	
	fbbh	15	5529	57	80	43	28	50	21	26	30	27	14	19	47	43	18	71	16	6	14	6	89.8%	10.2%	
	fcfc	16	80	5899	97	61	50	56	35	43	59	24	15	26	44	52	10	53	16	3	11	10	88.6%	11.4%	
	fejs	8	56	57	6163	57	22	43	26	32	43	24	21	33	33	33	16	25	10	9	10	19	91.4%	8.6%	
	ffmn	11	75	49	69	5539	36	50	42	56	44	50	13	37	36	28	21	40	30	18	22	14	88.2%	11.8%	
	fjam	13	43	56	48	44	3884	34	31	63	50	28	11	23	28	20	13	29	23	15	15	9	96.7%	13.3%	
	fmjd	3	28	20	55	21	22	5925	13	21	26	11	4	39	3	6	2	11	11	5	4	10	95.0%	5.0%	
	fsrb	13	63	47	71	84	33	43	5169	76	43	59	20	18	29	29	16	68	14	21	27	37	86.4%	13.6%	
	ftmj	22	40	24	60	47	43	42	35	5014	41	81	12	29	12	30	26	24	17	23	13	25	88.6%	11.4%	
	fwxs	16	66	49	85	50	32	44	19	33	4877	21	11	29	22	29	3	39	39	7	9	20	88.7%	11.3%	
	mbib	11	21	26	37	48	21	13	39	68	33	4882	20	21	24	31	33	38	11	45	15	23	89.4%	10.6%	
	mbiw	12	24	14	84	20	25	19	14	14	9	47	2386	33	49	29	27	22	41	11	27	18	24	81.7%	18.3%
	mbmg	6	27	15	50	15	11	55	10	12	17	37	27	5169	39	16	31	26	16	13	15	13	92.0%	8.0%	
	mcel	7	66	41	94	33	16	20	14	19	27	32	39	39	39	4106	51	13	75	18	21	17	12	86.3%	13.7%
	mcon	1	53	33	68	17	16	19	21	18	33	42	19	10	39	39	3865	8	80	21	12	16	9	87.3%	12.2%
mcfl	27	26	8	24	29	16	15	29	39	8	64	24	39	20	26	4808	21	8	56	12	21		90.4%	9.6%	
mrcb	1	56	47	57	34	18	18	20	13	41	41	21	18	54	54	8	3945	45	16	24	9		86.9%	13.1%	
majm	2	29	29	51	13	28	24	20	12	40	39	12	25	30	51	9	51	3527	8	19	1		87.7%	12.3%	
majr	55	7	5	12	14	5	12	13	15	4	52	19	19	33	18	30	19	2	4152	6	8		92.3%	7.7%	
masm	4	36	14	37	30	21	16	25	14	16	41	22	23	18	23	8	23	35	8	3593	11		89.4%	10.6%	
vineet	1	13	11	14	4		11	11	8	2	13	5	10	9	10	5	11	4	4	2	4384		96.7%	3.3%	
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>																									
<div><div>fash</div><div>fbbh</div><div>fcfc</div><div>fejs</div><div>ffmn</div><div>fjam</div><div>fmjd</div><div>fsrb</div><div>ftmj</div><div>fwxs</div><div>mbib</div><div>mbiw</div><div>mbmg</div><div>mcel</div><div>mcon</div><div>mcfl</div><div>mrcb</div><div>majm</div><div>majr</div><div>masm</div><div>vineet</div></div>																									
Predicted Class																									

		Test Accuracy (Per Frame)																						
True Class	fash	492	166	29	38	102	29	82	48	57	25	65	21	31	13	24	44	10	17	77	8	10	35.4%	64.6%
	fbbh	8	732	63	130	72	46	54	62	35	47	39	16	73	31	62	17	61	40	11	39	4	44.6%	55.4%
	folc	11	90	691	78	111	53	98	38	41	75	17	27	50	31	40	13	32	18	4	14	10	44.8%	55.2%
	fejs	5	82	84	765	70	16	44	19	36	22	11	37	95	21	23	13	33	9	4	15	18	53.8%	46.2%
	ffmm	28	154	117	157	665	77	139	97	168	87	86	30	74	39	47	44	28	30	24	13	18	31.3%	68.7%
	fjam	41	173	87	56	73	241	96	79	132	140	32	20	34	41	47	14	33	72	7	30	14	16.5%	83.5%
	fmjd	6	43	42	155	62	24	810	57	55	45	10	18	37	49	6	10	24	11	7	24	7	53.9%	46.1%
	fsrb	23	98	89	102	136	91	124	808	117	55	54	19	44	24	31	16	62	22	16	26	25	40.8%	59.2%
	ftmj	11	92	81	102	191	44	125	92	580	43	70	18	72	33	14	29	38	8	23	13	23	34.1%	65.9%
	fwxs	81	60	99	50	58	50	71	20	64	655	23	16	18	23	38	8	24	48	12	23	7	45.2%	54.8%
	mbib	12	56	27	34	76	42	25	27	84	49	618	56	51	55	67	36	41	15	74	29	28	41.1%	58.9%
	mbiw	15	26	28	74	41	25	39	24	28	29	113	366	32	136	42	21	67	13	53	30	20	30.0%	70.0%
	mbmg	4	49	69	55	66	31	100	36	81	43	95	50	854	78	54	87	59	17	42	20	32	44.4%	55.6%
	mcel	3	22	31	23	15	24	21	11	24	39	24	21	44	413	35	35	60	20	37	15	11	44.5%	55.5%
	mcoen	9	76	56	81	36	33	33	19	30	43	78	53	51	77	497	30	166	38	39	18	19	33.5%	66.5%
mcofl	24	51	32	54	51	40	38	34	75	24	88	58	118	110	42	700	35	10	94	19	25	40.7%	59.3%	
mrcb	5	37	58	58	48	23	38	41	48	27	108	68	46	126	68	15	829	67	27	24	21	46.5%	53.5%	
msjm	14	81	52	53	46	42	52	25	24	69	50	25	56	59	85	7	94	522	16	64	26	35.7%	64.3%	
msjr	24	11	17	25	38	14	17	18	27	14	116	47	79	143	42	66	85	18	621	7	13	43.1%	56.9%	
mann	13	71	37	57	41	34	37	24	63	37	88	51	67	57	51	19	95	75	17	996	18	51.1%	48.9%	
vineet	2	27	12	26	20	7	30	22	23	13	19	48	57	81	26	56	34	18	24	9	945	63.0%	37.0%	
		<div><div>89.2%</div><div>33.3%</div><div>38.4%</div><div>35.2%</div><div>33.0%</div><div>24.4%</div><div>39.1%</div><div>50.5%</div><div>32.4%</div><div>41.4%</div><div>34.3%</div><div>34.4%</div><div>43.1%</div><div>25.2%</div><div>37.1%</div><div>54.7%</div><div>43.4%</div><div>48.0%</div><div>50.5%</div><div>89.4%</div><div>73.0%</div></div>																						
		<div><div>40.8%</div><div>66.7%</div><div>61.6%</div><div>64.8%</div><div>67.0%</div><div>75.6%</div><div>60.9%</div><div>49.5%</div><div>67.8%</div><div>58.6%</div><div>65.7%</div><div>65.6%</div><div>56.9%</div><div>74.8%</div><div>62.9%</div><div>45.3%</div><div>56.6%</div><div>52.0%</div><div>49.5%</div><div>30.6%</div><div>27.0%</div></div>																						
		fash	fbbh	folc	fejs	ffmm	fjam	fmjd	fsrb	ftmj	fwxs	mbib	mbiw	mbmg	mcel	mcoen	mcofl	mrcb	msjm	msjr	mann	vineet		
		Predicted Class																						

Per audio file accuracy-

