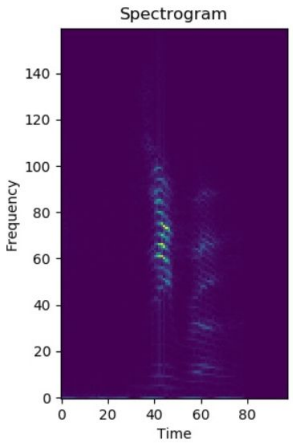
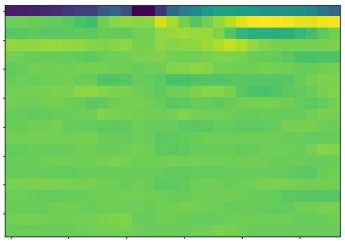
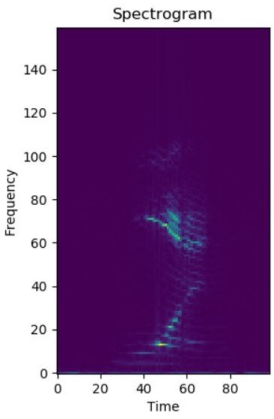
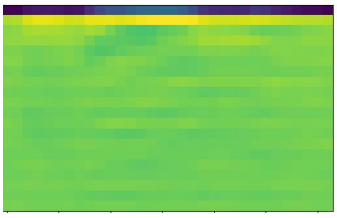
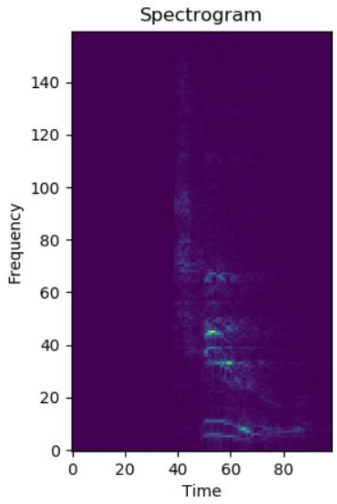
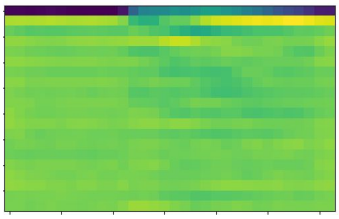
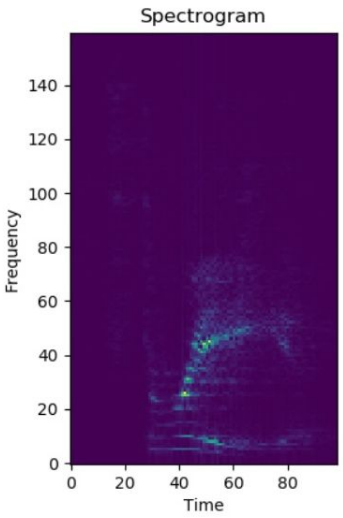
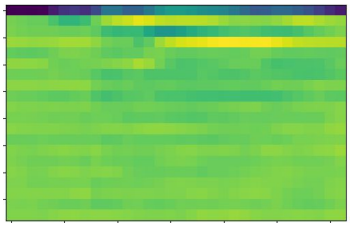
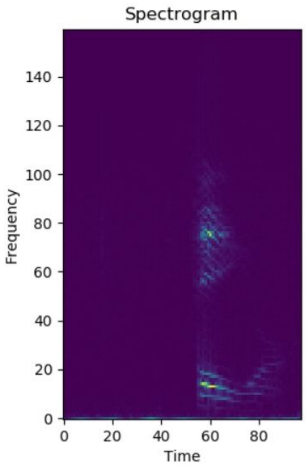
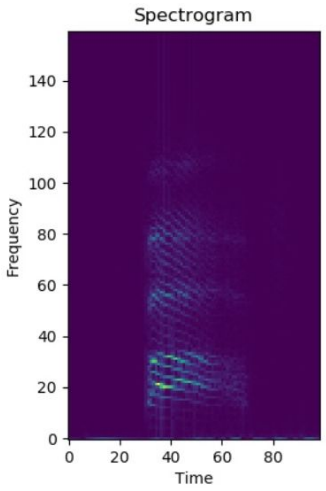
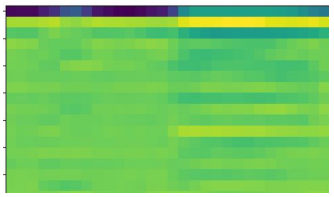
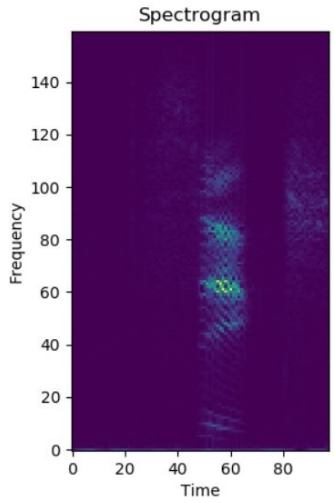
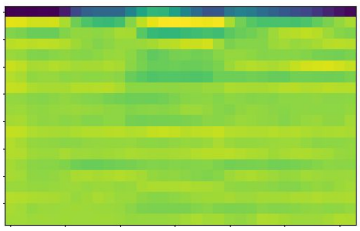
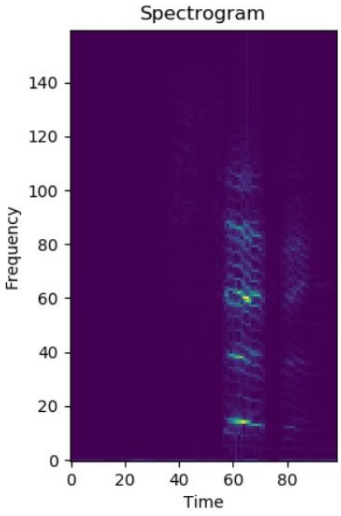
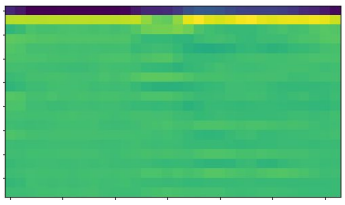
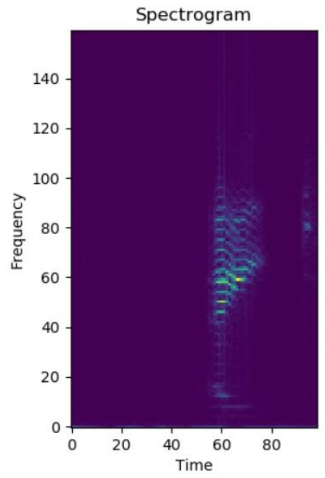
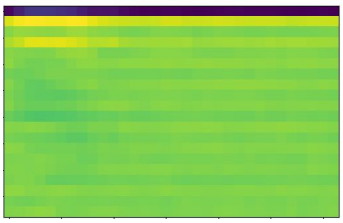
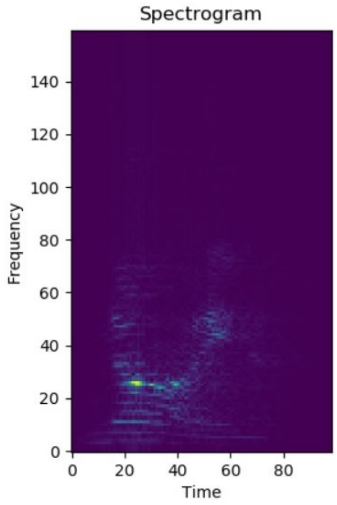
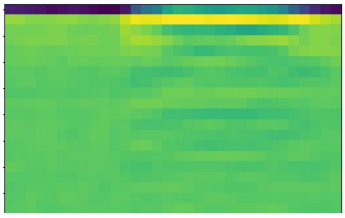
Assignment - 2
MCA-563
Abhishek Maiti
2016005

These are some of the plots for the features for different digits.

Digit	Spectrogram	MFCC
zero		
one		

two		
three		
four		

five		
six		
seven		

eight		
nine		

I used an SVM with an RBF kernel and a LibSVM optimiser. These are the results that I got with and without augmentations. For augmentations, due to less compute power I randomly chose a noise from the 6 available and added that to the sample.

I added the noise to the test set to see how the noise affects the accuracy, due to the huge number of training samples, I couldn't add all noises to all the training samples hence couldn't make the features that robust.

Validation Accuracy:

	Without Augmentation	**With Augmentation**
MFCC	45%	22%
Spectrogram	63%	32%

Precision-Recall Table MFCC without Augmentation

	precision	**recall**	**f1-score**
0	0.85	0.58	0.69
1	0.31	0.71	0.43
2	0.38	0.54	0.45
3	0.47	0.63	0.54
4	0.94	0.11	0.20
5	0.63	0.23	0.33
6	0.83	0.04	0.07
7	0.40	0.75	0.52
8	0.44	0.52	0.48
9	0.55	0.50	0.52
micro avg	**0.45**	**0.45**	**0.45**
macro avg	**0.58**	**0.46**	**0.42**
weighted avg	**0.59**	**0.45**	**0.42**

Precision-Recall Table MFCC tested with Augmentation testing samples

	precision	**recall**	**f1-score**
0	0.23	0.29	0.24
1	0.25	0.27	0.26
2	0.19	0.17	0.17
3	0.22	0.29	0.23
4	0.18	0.24	0.32
5	0.27	0.23	0.24

6	0.24	0.21	0.22
7	0.29	0.22	0.24
8	0.14	0.21	0.17
9	0.25	0.25	0.25
micro avg	**0.22**	**0.26**	**0.24**
macro avg	**0.24**	**0.27**	**0.25**
weighted avg	**0.25**	**0.29**	**0.26**	**2494**

Precision-Recall Table Spectrogram without Augmentation

	precision	**recall**	**f1-score**
0	0.59	0.57	0.58
1	0.48	0.63	0.54
2	0.44	0.53	0.48
3	0.53	0.63	0.57
4	0.78	0.76	0.77
5	0.65	0.56	0.60
6	0.79	0.74	0.76
7	0.72	0.55	0.63
8	0.73	0.62	0.67
9	0.60	0.57	0.59
micro avg	**0.62**	**0.62**	**0.62**
macro avg	**0.63**	**0.62**	**0.62**
weighted avg	**0.63**	**0.62**	**0.62**

Precision-Recall Table Spectrogram tested on Augmentated test samples

	precision	**recall**	**f1-score**
0	0.32	0.30	0.31
1	0.31	0.29	0.29
2	0.32	0.29	0.31
3	0.39	0.32	0.34
4	0.20	0.18	0.19
5	0.24	0.24	0.24
6	0.33	0.30	0.31
7	0.23	0.20	0.22
8	0.25	0.21	0.23
9	0.30	0.27	0.29
micro avg	**0.32**	**0.34**	**0.33**
macro avg	**0.33**	**0.35**	**0.33**

Result: We can see that, if we add distortion, the classifier isn't able to detect properly hence we see a huge drop of accuracy. We don't see any particular drop for any number and all digits more or less dropped to the same extent. I was expecting something of the sort which is seen on image dataset when the classifier confuses between 3 and 8 but that wasn't the case.