Assignment - 2

Priyanshi Jain
2017358

Part 1 (Spectrogram) :

For computing spectrogram, the given audio is divided into some windows and a fourier transform is taken to convert into frequency domain.
Window size of 20ms is taken and an overlap of 50% is considered between adjacent windows.

SVM Results:

	precision	recall	f1-score	support
0	0.95	0.77	0.85	260
1	0.65	0.78	0.71	230
2	0.57	0.73	0.64	236
3	0.73	0.74	0.73	248
4	0.71	0.65	0.68	280
5	0.66	0.47	0.55	242
6	0.86	0.78	0.82	262
7	0.84	0.74	0.79	263
8	0.64	0.82	0.72	243
9	0.68	0.71	0.70	230
accuracy			0.72	2494
macro avg	0.73	0.72	0.72	2494
weighted avg	0.73	0.72	0.72	2494

Part 2 (MFCC):

Steps for computing MFCC:

- After loading the audio, a pre-emphasis filter is applied. (coefficient = 0.97)
- Then, similar to spectrogram, the audio is divided into windows and features are computed.
- After that, using those filters, filter banks are computed.
- DCT of filter banks is taken to give MFCCs.

SVM results:

	precision	recall	f1-score	support
0	0.91	0.91	0.91	260
1	0.88	0.84	0.86	230
2	0.77	0.82	0.79	236
3	0.88	0.85	0.86	248
4	0.91	0.89	0.90	280
5	0.87	0.81	0.84	242
6	0.87	0.91	0.89	262
7	0.83	0.87	0.85	263
8	0.84	0.92	0.88	243
9	0.90	0.81	0.85	230
accuracy			0.86	2494
macro avg	0.87	0.86	0.86	2494
weighted avg	0.87	0.86	0.86	2494

Part 3:

Data Augmentation:

For data augmentation, background noises were added to 40% of the training data. These background noises were chosen randomly.

Spectrogram results:

	precision	recall	f1-score	support
0	0.92	0.75	0.83	260
1	0.58	0.77	0.66	230
2	0.48	0.71	0.57	236
3	0.73	0.66	0.69	248
4	0.69	0.55	0.61	280
5	0.63	0.43	0.51	242
6	0.88	0.71	0.78	262
7	0.80	0.72	0.76	263
8	0.57	0.79	0.66	243
9	0.65	0.67	0.66	230
accuracy			0.67	2494
macro avg	0.69	0.67	0.67	2494
weighted avg	0.70	0.67	0.67	2494

MFCC results:

	precision	recall	f1-score	support
0	0.88	0.92	0.90	260
1	0.87	0.84	0.85	230
2	0.82	0.81	0.81	236
3	0.85	0.85	0.85	248
4	0.90	0.88	0.89	280
5	0.84	0.80	0.82	242
6	0.86	0.90	0.88	262
7	0.85	0.84	0.84	263
8	0.83	0.91	0.87	243
9	0.88	0.81	0.84	230
accuracy			0.86	2494
macro avg	0.86	0.85	0.86	2494
weighted avg	0.86	0.86	0.86	2494

It can be seen that the results are good, which shows the robustness of the classifier.
Also, MFCC results are better than those from spectrogram.

References for spectrogram:

- https://fairyonice.github.io/implement-the-spectrogram-from-scratch-in-python.html
- https://towardsdatascience.com/understanding-audio-data-fourier-transform-fft-spectrogram-and-speech-recognition-a4072d228520
- http://www.frank-zalkow.de/en/code-snippets/create-audio-spectrograms-with-python.html?i=1

References for MFCC:

- https://github.com/jameslyons/python_speech_features/blob/9a2d76c6336d969d51ad3aa0d129b99297dcf55e/python_speech_features/base.py#L179
- https://haythamfayek.com/2016/04/21/speech-processing-for-machine-learning.html