MCA Assignment 2

Suryatej Reddy (2016102)

March 2020

1 Answer 1

The code for creating a spectogram is attached along with the submission. Some spectograms are attached below

1.1 Zero

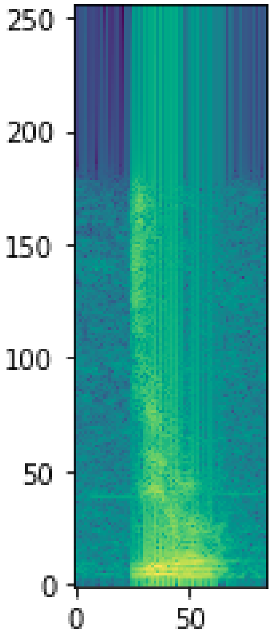


Figure 1: Spectogram of the Digit Zero

1.2 Four

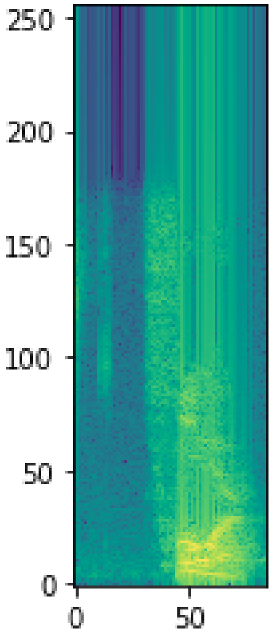


Figure 2: Spectogram of the Digit Four

2 Answer 2

The code for generating MFCC is attached along with the submission. Some MFCC visualizations are attached below

2.1 Zero

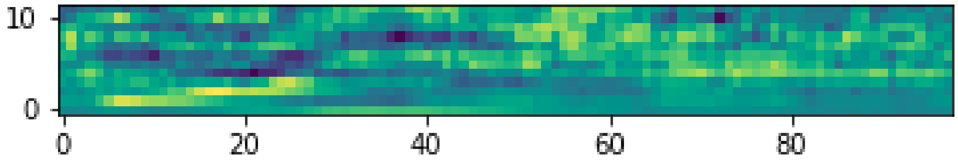


Figure 3: MFCC of the Digit Zero

2.2 Four

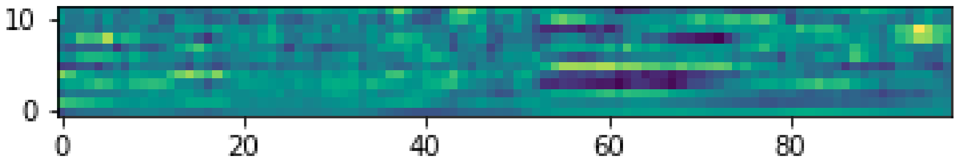


Figure 4: MFCC of the Digit Four

3 Answer 3

3.1 Models

Small amount of noise was added to each audio in the training sample. The exact noise file being added was selected randomly.

For identifying the best models, I performed a grid search on the validation set with different parameters. The grid search objects are attached along with the submission.

The best parameters over which the training data were fit are:

- **Spectogram** - Linear Kernel; C = 1
- **MFCC** - Linear Kernel; C = 0.3

3.2 Results

The values reported are **micro** scale.

3.2.1 Training Set

Model	**Accuracy**	**Precision**	**Recall**	**FScore**
Spectogram Without Noise	1	1	1	1
Spectogram With Noise in Training Set	1	1	1	1
MFCC Wihout Noise	1	1	1	1
Spectogram With Noise in Training Set	1	1	1	1

3.2.2 Validation Set

Model	Accuracy	Precision	Recall	FScore
Spectogram Without Noise	0.509	0.509	0.509	0.509
Spectogram With Noise in Training Set	0.517	0.517	0.517	0.517
MFCC Wihout Noise	0.390	0.390	0.390	0.390
Spectogram With Noise in Training Set	0.397	0.397	0.397	0.397

4 Analysis

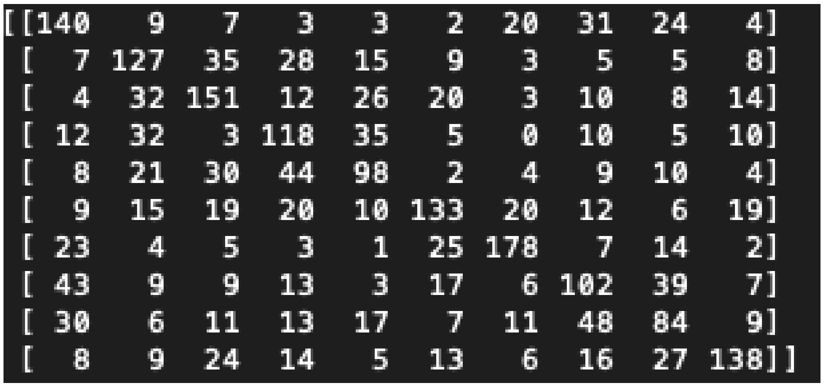
In both cases (Spectogram and MFCC), the model trained with noise augmention works better than their vanilla counterpart. This is as expected because the added noise makes the model more robust.

The best performing model was the spectogram that was trained with noise augmented data and had an accuracy/precision of **0.52**. The model accuracy could have been improved further by performing grid search on a wider range of values than I already have.

The block size I chose for creating the spectogram was 512 with 50% overlap in blocks. I did not have the resources for attempting multiple overlap values but this is another factor that would have improved the performance. I realized that in shorter audios, by limiting the overlap, the spectograms could be formed better, because it's only a single word so overlap wouldn't be required. This observation is based on manually generating spectograms with different values.

4.1 Spectogram

4.1.1 Classification Without Noise

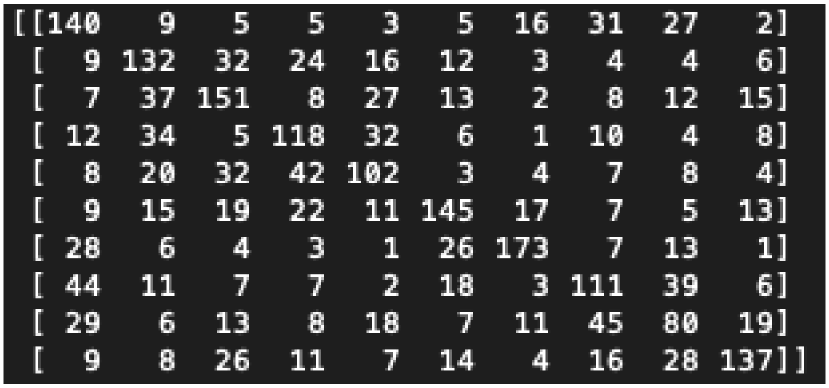


```
[[140   9   7   3   3   2  20  31  24   4]
 [  7 127  35  28  15   9   3   5   5   8]
 [  4  32 151  12  26  20   3  10   8  14]
 [ 12  32   3 118  35   5   0  10   5  10]
 [  8  21  30  44  98   2   4   9  10   4]
 [  9  15  19  20  10 133  20  12   6  19]
 [ 23   4   5   3   1  25 178   7  14   2]
 [ 43   9   9  13   3  17   6 102  39   7]
 [ 30   6  11  13  17   7  11  48  84   9]
 [  8   9  24  14   5  13   6  16  27 138]]
```

We can see from the confusion matrix, the classification of digits 4 and 9 do not perform well.

- 4 is often confused with 3 and 2.
- 9 is confused with 8 and 0.

4.1.2 Classification With Noise



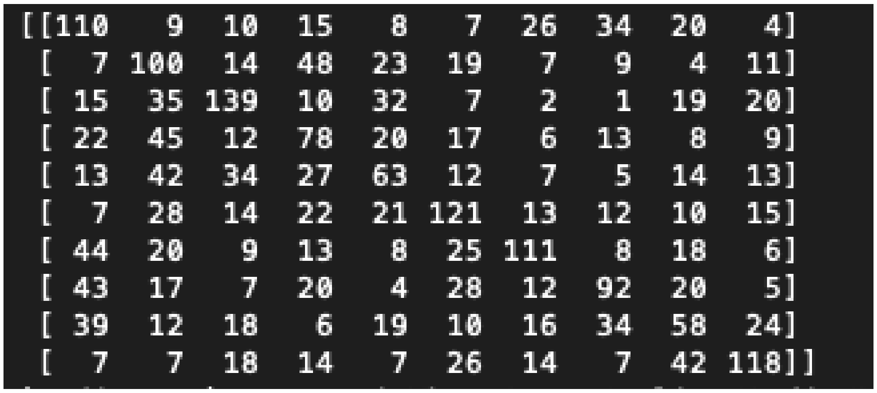
```
[[140   9   5   5   3   5  16  31  27   2]
 [  9 132  32  24  16  12   3   4   4   6]
 [  7  37 151   8  27  13   2   8  12  15]
 [ 12  34   5 118  32   6   1  10   4   8]
 [  8  20  32  42 102   3   4   7   8   4]
 [  9  15  19  22  11 145  17   7   5  13]
 [ 28   6   4   3   1  26 173   7  13   1]
 [ 44  11   7   7   2  18   3 111  39   6]
 [ 29   6  13   8  18   7  11  45  80  19]
 [  9   8  26  11   7  14   4  16  28 137]]
```

This is the best performing model and we can see that the classification performance for both 4 and 9 has significantly increased in this case.

4.2 MFCC

MFCC with noise works slightly better than without noise on the validation set. This is exactly as we expect because the noise makes the model more robust.

4.2.1 Classification With Noise

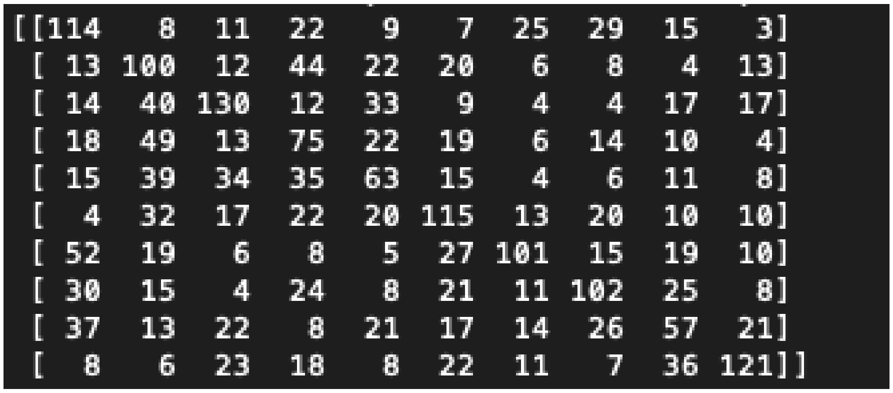


```
[[110   9  10  15   8   7  26  34  20   4]
 [  7 100  14  48  23  19   7   9   4  11]
 [ 15  35 139  10  32   7   2   1  19  20]
 [ 22  45  12  78  20  17   6  13   8   9]
 [ 13  42  34  27  63  12   7   5  14  13]
 [  7  28  14  22  21 121  13  12  10  15]
 [ 44  20   9  13   8  25 111   8  18   6]
 [ 43  17   7  20   4  28  12  92  20   5]
 [ 39  12  18   6  19  10  16  34  58  24]
 [  7   7  18  14   7  26  14   7  42 118]]
```

As we can see from the confusion matrix, the digits 3,4 and 8 do not perform well.

- 8 has the worst classification. As we can see, 8 is confused a lot with zero and seven.
- Next worst performing is 4 and majority of 4 is confused with 2.
- Third worst is 3 and majority of 3 is confused with 1.

4.2.2 Classification Without Noise



```
[[114   8  11  22   9   7  25  29  15   3]
 [ 13 100  12  44  22  20   6   8   4  13]
 [ 14  40 130  12  33   9   4   4  17  17]
 [ 18  49  13  75  22  19   6  14  10   4]
 [ 15  39  34  35  63  15   4   6  11   8]
 [  4  32  17  22  20 115  13  20  10  10]
 [ 52  19   6   8   5  27 101  15  19  10]
 [ 30  15   4  24   8  21  11 102  25   8]
 [ 37  13  22   8  21  17  14  26  57  21]
 [  8   6  23  18   8  22  11   7  36 121]]
```

Even in this case, we can see that the same digits 3,4 and 8 do not perform well and the behavior is similar.

5 Files

https://drive.google.com/drive/folders/1396XFtyT7dMhAz9bjza3rAN-Sj8-2eW_?usp=sharing

https://drive.google.com/drive/folders/1iMSVC_pGbEJ0j3sZPViPyMcl0usTskQm?usp=sharing