Weekly Report

Wangwon Lee, 2019/03/30

This week	Next week
• **Classification Report** - Confusion matrix - Precision , Recall, F1-score • **Visualization** - Filter and Activation map - Frequency response - Spectrogram	• **Fine Tuning** - Change activation function - Change filter size - Try other reference • **Visualization** - Each label - The other model - CAM(Class Activation Map)

Interesting and new finding

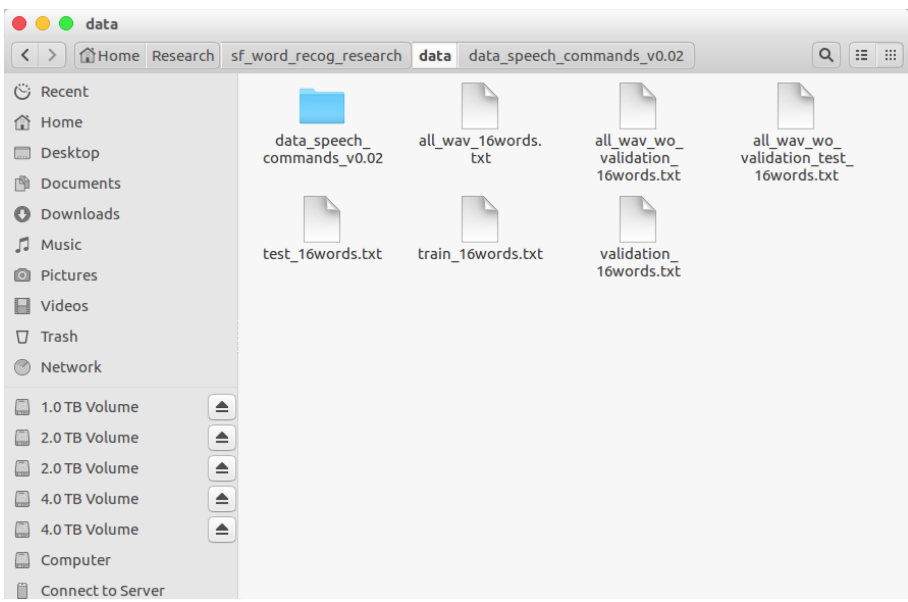
- Fine Tuning
- Visualization

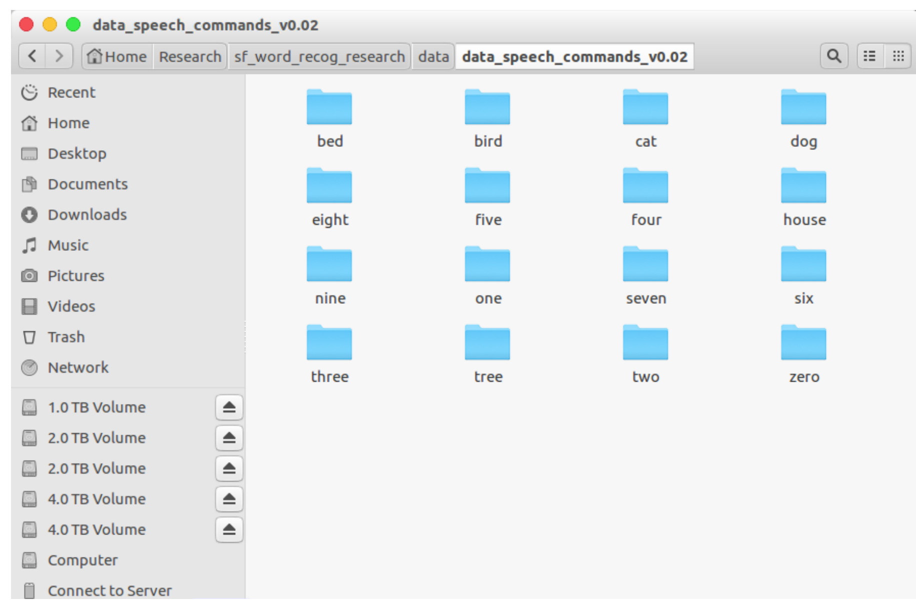
The aim of this month / Discussion

The aim of this month: To study the brain and GLM, To investigate about CNN.

Audio Classification - Previous Work

- Data is low-waveform.
  - sec: 1, sampling rate: 16000, type: float32, channel: mono
- 16 class data.
  - 'zero', 'one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'bed', 'bird', 'tree', 'cat', 'house', 'dog'
- Train: 40851(≒80%), Validation: 4796(≒10%), Test: 5297(≒10%)

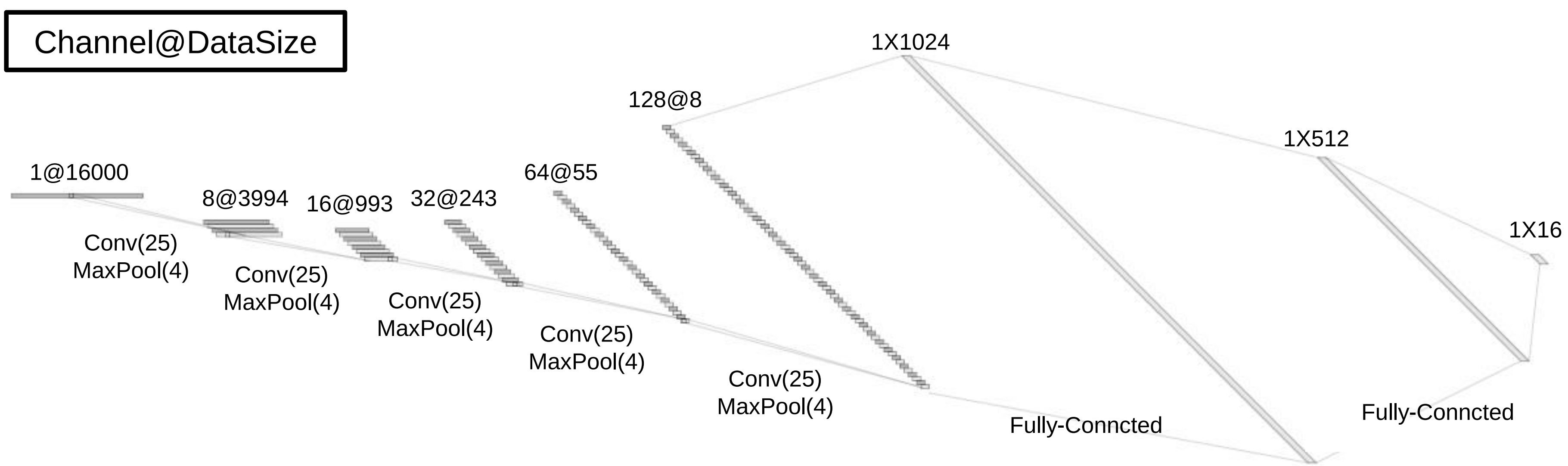




Audio Classification - Previous Work

- For example, ‘5Conv, 2FC’ baseline model’s detail.
- It just flatten 2D model. (5X5 filter->1X25 filter, 2X2 stride->1X4 stride)

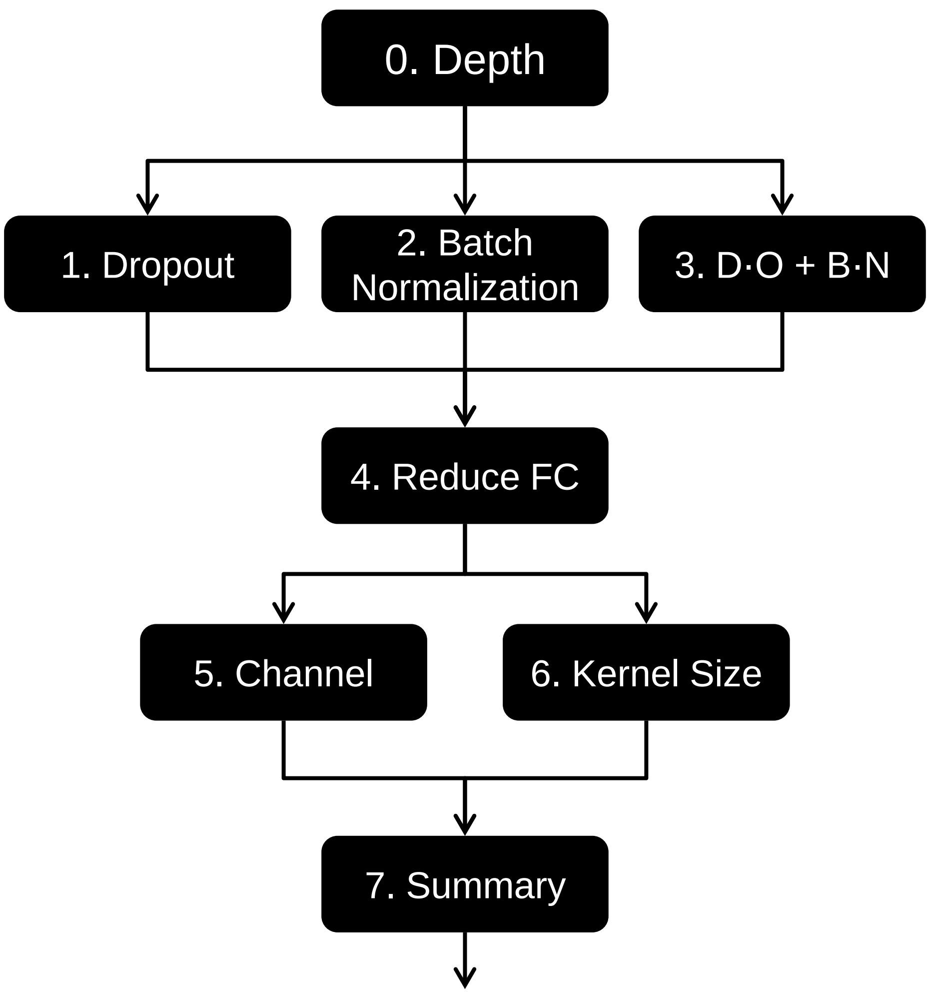
- Input: 16000X1 low waveform.
- Output:1x16 labeled one hot vector. ('zero', …, 'eight', …, 'house', 'dog')
- Loss: cross entropy loss
- Obtimizer: Adam

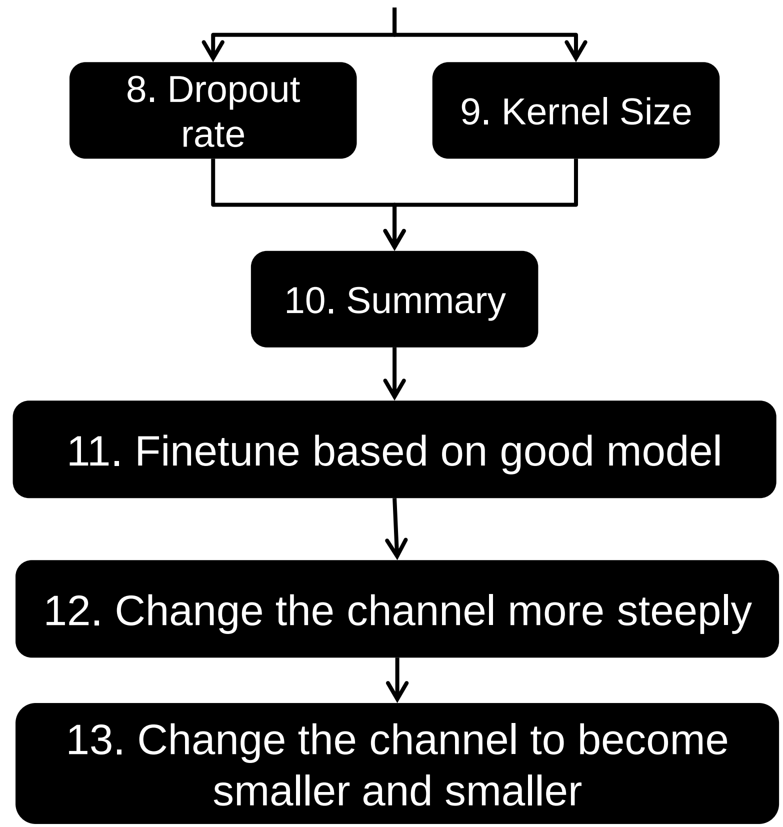


[Baseline Architecture]

Audio Classification - Previous Work

- Fine tuning task in 1D-CNN





Previous Work

Audio Classification

- This is SOTA(State Of The Art) in current research.

	Architecture (i = 0,1,2…)	1D DO(0.5)	1D BN	1D DO+BN	Params
	baseline				
base model	5 Conv(25, $8*2^i$), 5 Pool(4), 2 FC	0.9090	0.9072	0.9240	1,855,056
	Accuracy and Number of parameters				
Custom channel 32 DO(0.75)	8 CONV(5, 64)	0.9533	0.9285	0.9391	94,768
Custom channel 64 DO(0.75)	8 CONV(5, 128)	0.9589	X	0.9497	363,600
Custom VGG style DO(0.75)	16 CONV(3, 128) , 8 Pool	0.9620	0.9423	0.9136	470,736
	Only Accuracy				
Custom channel 128 DO(0.25)+BN	9 CONV(5, 512)	0.9535	X	0.9701	2,071,184

Audio Classification

- Confusion matrix
- Compare two best model.
- Not much different, but the right model is a little better.

Actual class

Predict Class

Zero	[[369	0	5	1	3	0	1	4	0	1	0	0	1	0	0	0]
One	[1	347	0	0	4	1	1	0	0	7	2	0	0	1	0	0]
Two	[8	0	369	2	0	0	0	0	1	0	0	0	2	2	0	0]
Three	[1	0	2	356	0	1	1	2	6	0	0	0	0	0	0	8]
Fore	[2	2	1	1	359	2	0	0	0	0	0	0	1	0	0	0]
Five	[0	6	0	3	4	386	0	2	1	3	0	1	2	0	0	0]
Six	[0	0	0	3	1	0	366	1	2	0	1	0	0	0	0	0]
Seven	[3	0	0	0	1	0	2	367	0	0	3	0	0	0	0	0]
Eight	[1	0	1	2	1	0	0	0	367	0	2	0	1	0	0	1]
Nine	[0	6	0	0	0	3	0	0	0	364	3	1	0	0	0	0]
Bed	[1	0	2	0	0	0	0	0	6	0	166	5	2	1	0	0]
Bird	[0	1	1	1	0	0	2	0	0	2	7	137	0	2	0	0]
Cat	[0	0	0	0	0	0	0	1	0	0	1	0	161	4	1	0]
Dog	[0	1	5	0	0	0	0	0	0	1	1	0	2	182	0	0]
House	[0	0	2	0	0	1	1	0	0	1	0	0	5	1	156	0]
Tree	[2	0	2	15	0	0	1	0	4	1	0	0	0	0	0	138]]

Custom channel 32 DO(0.75) **(Acc: 0.9533)**

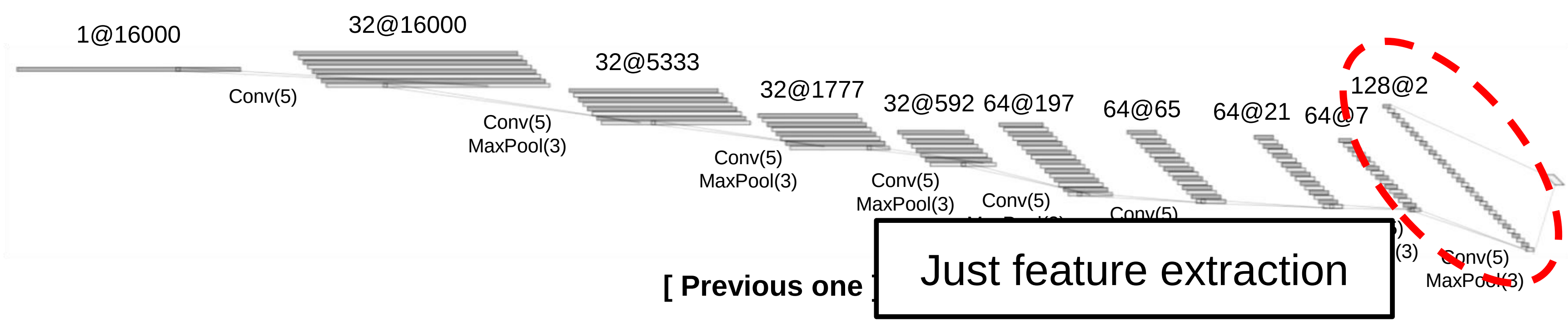
Actual class

[[375	0	3	0	2	0	0	3	0	1	0	0	0	0	0	1]
[0	348	0	0	4	1	0	1	1	6	1	0	0	0	1	1]
[1	0	375	2	1	0	0	0	0	0	0	0	0	2	1	2]
[1	0	5	356	0	0	2	3	1	0	1	0	0	0	0	8]
[0	0	0	0	364	1	0	0	0	0	0	1	1	0	0	1]
[0	0	0	0	5	399	0	0	1	2	0	0	1	0	0	0]
[0	0	0	1	0	0	371	1	0	0	1	0	0	0	0	0]
[1	0	0	0	0	0	1	373	1	0	0	0	0	0	0	0]
[0	0	4	2	0	0	0	0	362	4	0	2	1	0	0	1]
[0	5	0	0	0	3	0	1	0	363	2	3	0	0	0	0]
[0	0	1	0	0	0	1	0	1	0	178	1	0	0	0	1]
[0	0	0	0	0	0	0	0	0	1	3	147	0	1	0	1]
[0	0	0	0	0	0	0	0	0	0	0	0	166	1	1	0]
[0	0	2	0	0	0	0	0	0	0	1	1	0	186	0	2]
[0	0	0	0	0	0	0	0	2	0	0	1	3	0	161	0]
[0	0	1	10	0	0	0	0	4	1	0	0	0	0	0	147]]

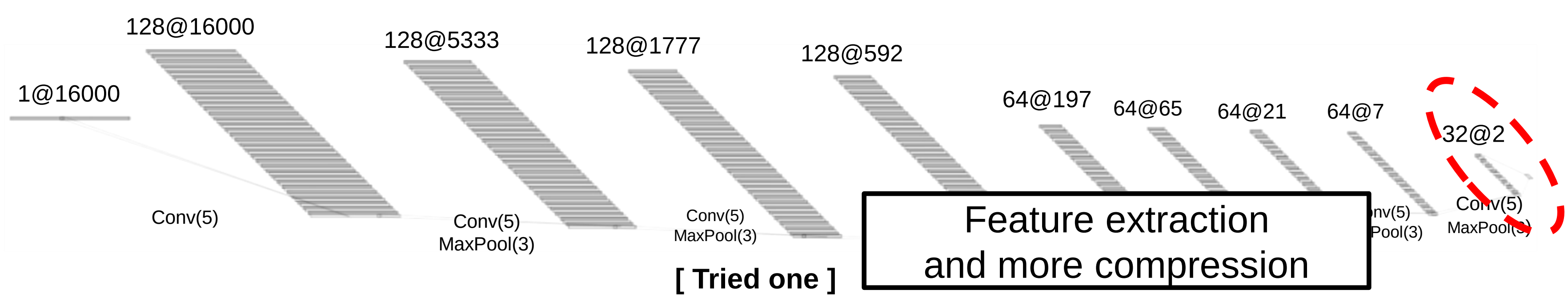
Custom channel 128 DO(0.25)+BN **(Acc: 0.9701)**

Audio Classification

- I think this try is very meaningful.
- Because this model focus on the feature compression than previous model.
- Like autoencoder. (16000 → 64(=2*32channel))
- This model is better when use other classifiers after extract the feature from CNN.



[Previous one]



[Tried one]

Audio Classification

- In summary, This is SOTA(State Of The Art) in this research.

- Though the performance is lower than the previous model, I think it is meaningful.
- The original data size, 16000, was compressed to 32 and the performance was 0.9327.
- And the other model compress to 128 and the performance was 0.9628.

- What if we use a feature from CNN as input to another model, this model will be very useful.

	Architecture (i = 0,1,2…)	1D DO(0.5)	1D BN	1D DO+BN	Params
	16000(Input data size) → 2(Length) * 16(Channel) = 32 (Feature size)				
Feature extraction ch 64	9 CONV(5, 16)	0.9238	0.9074	0.9327	91,712
	16000(Input data size) → 2(Length) * 64(Channel) = 128 (Feature Size)				
Feature extraction ch 256	9 CONV(5, 64)	0.9603	0.9572	0.9628	1,438,544

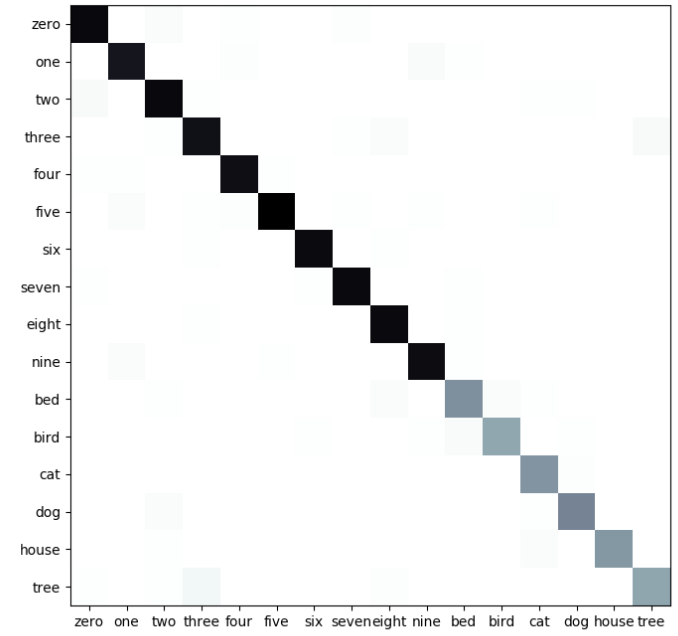
Audio Classification

- In previous time, I shown only accuracy and confusion matrix.
- But there is a lot of way that I show the model's performance more efficiently
- First, Visualize the number.

Actual class

Predict Class

Zero	[[369	0	5	1	3	0	1	4	0	1	0	0	1	0	0	0]
One	[1	347	0	0	4	1	1	0	0	7	2	0	0	1	0	0]
Two	[8	0	369	2	0	0	0	0	1	0	0	0	2	2	0	0]
Three	[1	0	2	356	0	1	1	2	6	0	0	0	0	0	0	8]
Fore	[2	2	1	1	359	2	0	0	0	0	0	0	1	0	0	0]
Five	[0	6	0	3	4	386	0	2	1	3	0	1	2	0	0	0]
Six	[0	0	0	3	1	0	366	1	2	0	1	0	0	0	0	0]
Seven	[3	0	0	0	1	0	2	367	0	0	3	0	0	0	0	0]
Eight	[1	0	1	2	1	0	0	0	367	0	2	0	1	0	0	1]
Nine	[0	6	0	0	0	3	0	0	0	364	3	1	0	0	0	0]
Bed	[1	0	2	0	0	0	0	0	6	0	166	5	2	1	0	0]
Bird	[0	1	1	1	0	0	2	0	0	2	7	137	0	2	0	0]
Cat	[0	0	0	0	0	0	0	1	0	0	1	0	161	4	1	0]
Dog	[0	1	5	0	0	0	0	0	0	1	1	0	2	182	0	0]
House	[0	0	2	0	0	1	1	0	0	1	0	0	5	1	156	0]
Tree	[2	0	2	15	0	0	1	0	4	1	0	0	0	0	0	138]]



Custom channel 32 DO(0.75) **(Acc: 0.9533)**

Audio Classification

- To validate the model, We can use measure such as precision, recall, f1-score.
- We find out that It does not confuse between ‘three’ and ‘tree’.
- It just confuses ‘tree’ as ‘three’.
- And It also confuse between ‘Bed’ and ‘Bird’.

Actual class

Predict Class

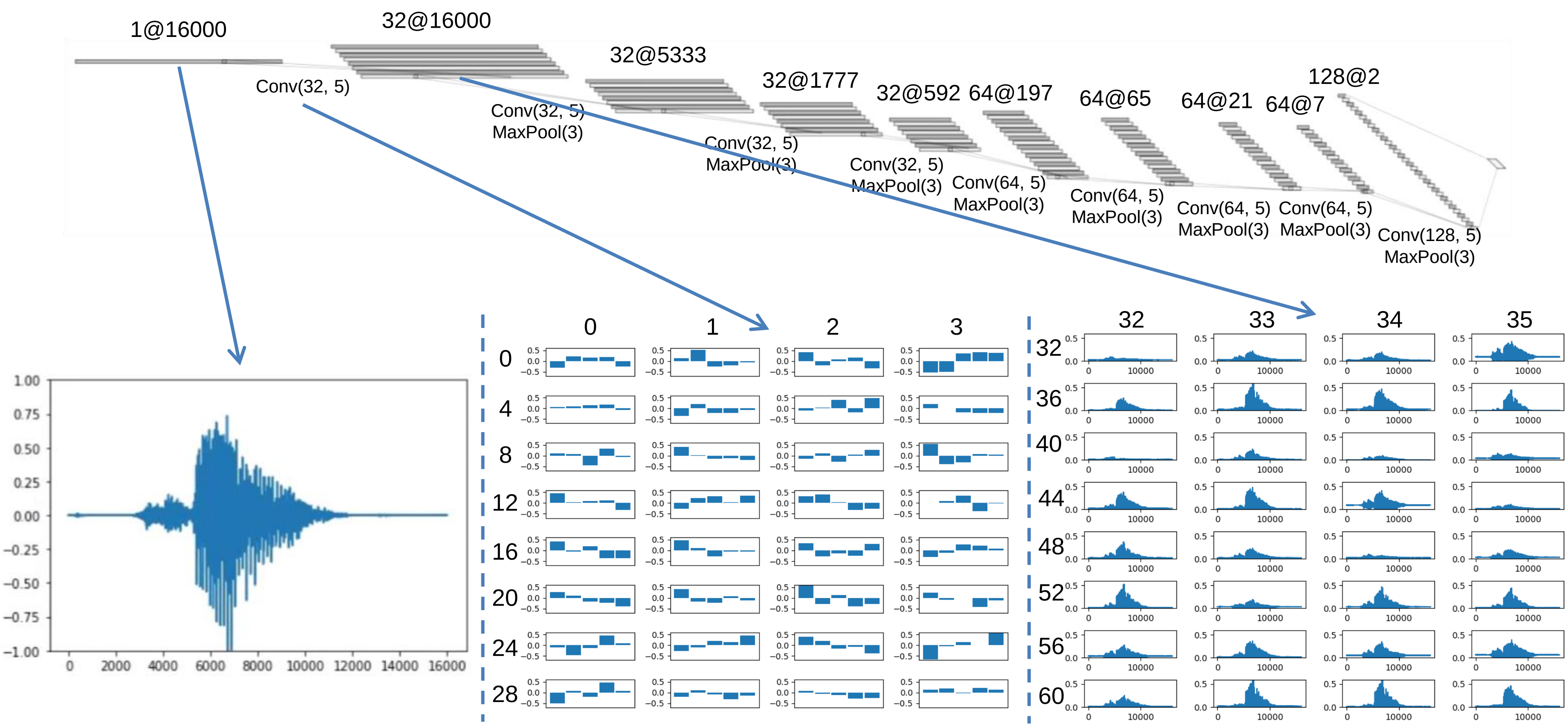
Zero	[[369	0	5	1	3	0	1	4	0	1	0	0	1	0	0	0]
One	[1	347	0	0	4	1	1	0	0	7	2	0	0	1	0	0]
Two	[8	0	369	2	0	0	0	0	1	0	0	0	2	2	0	0]
Three	[1	0	2	356	0	1	1	2	6	0	0	0	0	0	0	8]
Fore	[2	2	1	1	359	2	0	0	0	0	0	0	1	0	0	0]
Five	[0	6	0	3	4	386	0	2	1	3	0	1	2	0	0	0]
Six	[0	0	0	3	1	0	366	1	2	0	1	0	0	0	0	0]
Seven	[3	0	0	0	1	0	2	367	0	0	3	0	0	0	0	0]
Eight	[1	0	1	2	1	0	0	0	367	0	2	0	1	0	0	1]
Nine	[0	6	0	0	0	3	0	0	0	364	3	1	0	0	0	0]
Bed	[1	0	2	0	0	0	0	0	6	0	166	5	2	1	0	0]
Bird	[0	1	1	1	0	0	2	0	0	2	7	137	0	2	0	0]
Cat	[0	0	0	0	0	0	0	1	0	0	1	0	161	4	1	0]
Dog	[0	1	5	0	0	0	0	0	0	1	1	0	2	182	0	0]
House	[0	0	2	0	0	1	1	0	0	1	0	0	5	1	156	0]
Tree	[2	0	2	15	0	0	1	0	4	1	0	0	0	0	0	138]]

Custom channel 32 DO(0.75) **(Acc: 0.9533)**

	Precision	Recall	F1-score	Support
zero	0.95	0.96	0.95	385
one	0.96	0.95	0.95	364
two	0.95	0.96	0.95	384
three	0.93	0.94	0.94	377
four	0.96	0.98	0.97	368
five	0.98	0.95	0.96	408
six	0.98	0.98	0.98	374
seven	0.97	0.98	0.97	376
eight	0.95	0.98	0.96	376
nine	0.96	0.97	0.96	377
bed	0.89	0.91	0.90	183
bird	0.95	0.90	0.92	153
cat	0.91	0.96	0.93	168
dog	0.94	0.95	0.95	192
house	0.99	0.93	0.96	167
tree	0.94	0.85	0.89	163
weighted avg	0.95	0.95	0.95	4815

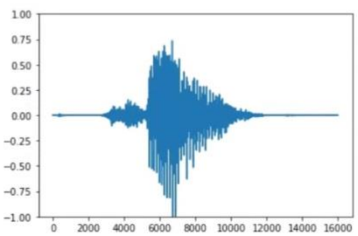
Audio Classification

- Visualize the filter map. (Custom channel 32 DO(0.75) Model)
- Less than zero of the waveform is removed because of ‘Relu’
- Because of this problem, it is difficult to analysis from the point of view of Signal Processing.
- So, I consider to use ‘tanh’ function.

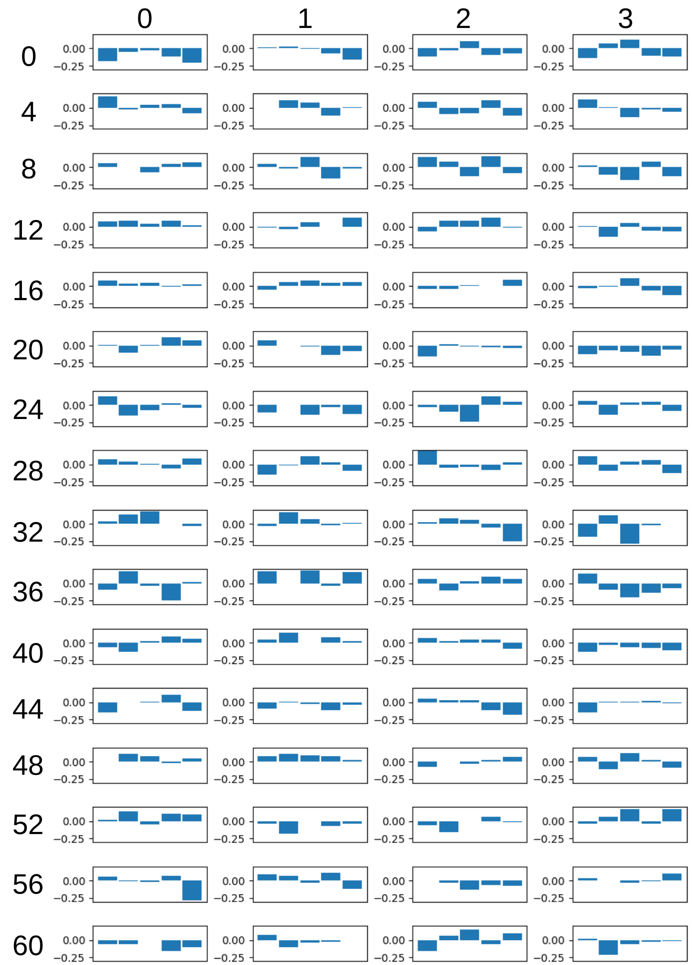


Audio Classification

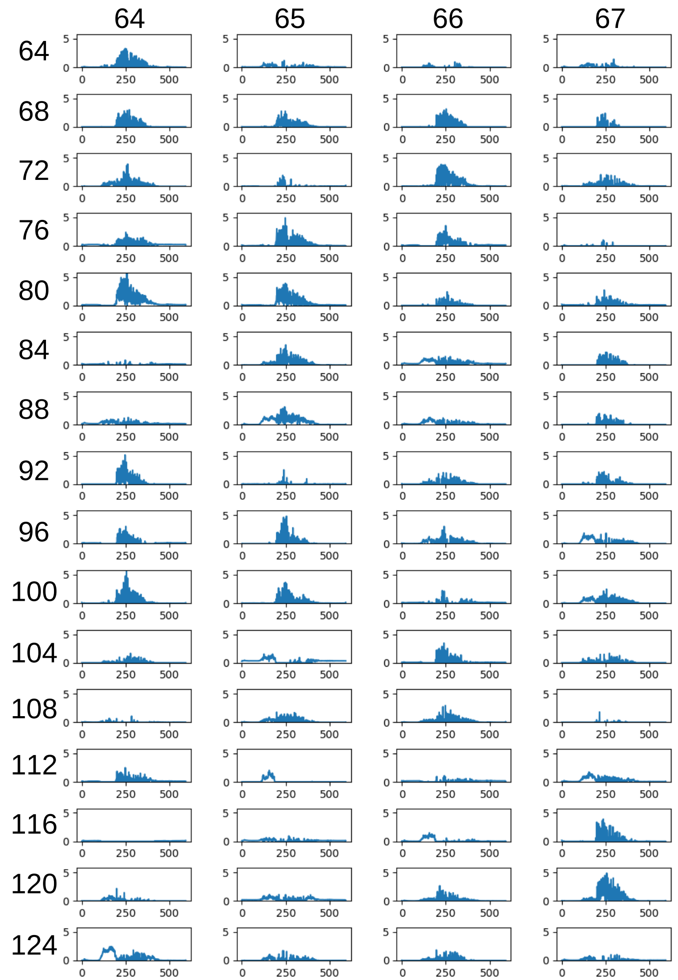
- Visualize the filter map. (Custom channel 32 DO(0.75) Model)
- Going through the filter, the information is disappearing more and more.



Original input



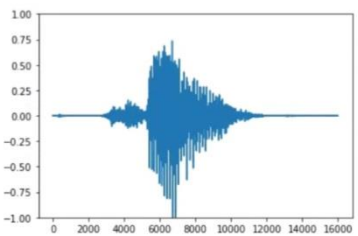
5th Layer - Filter



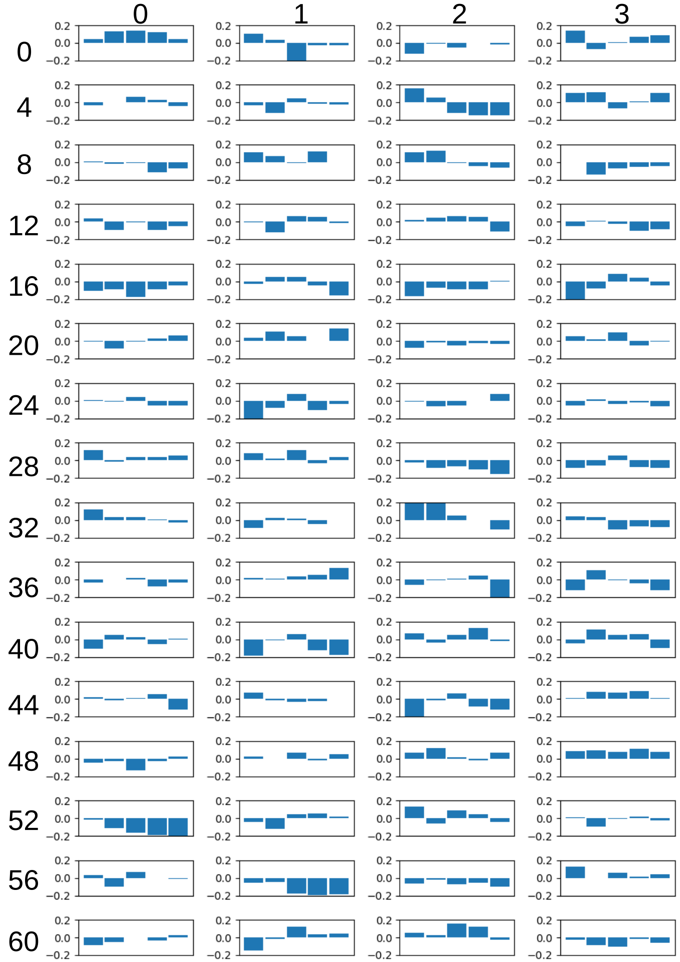
5th Layer – Activation map

Audio Classification

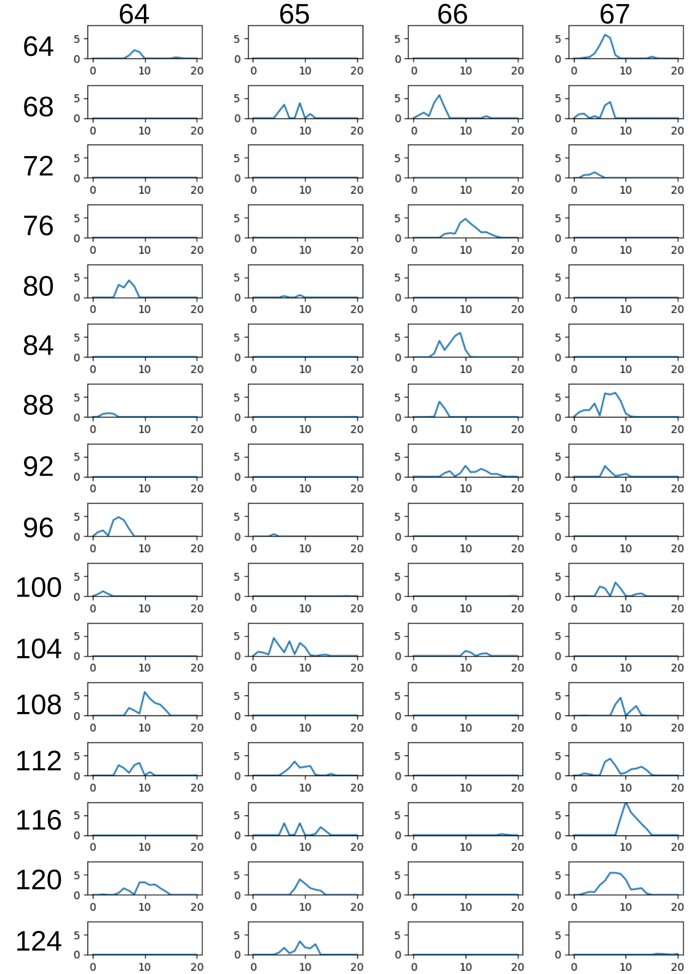
- About half is gone away...
- I don't know whether each label has a place for feature extraction or it is originally empty



Original input



8th Layer - Filter



8th Layer – Activation map

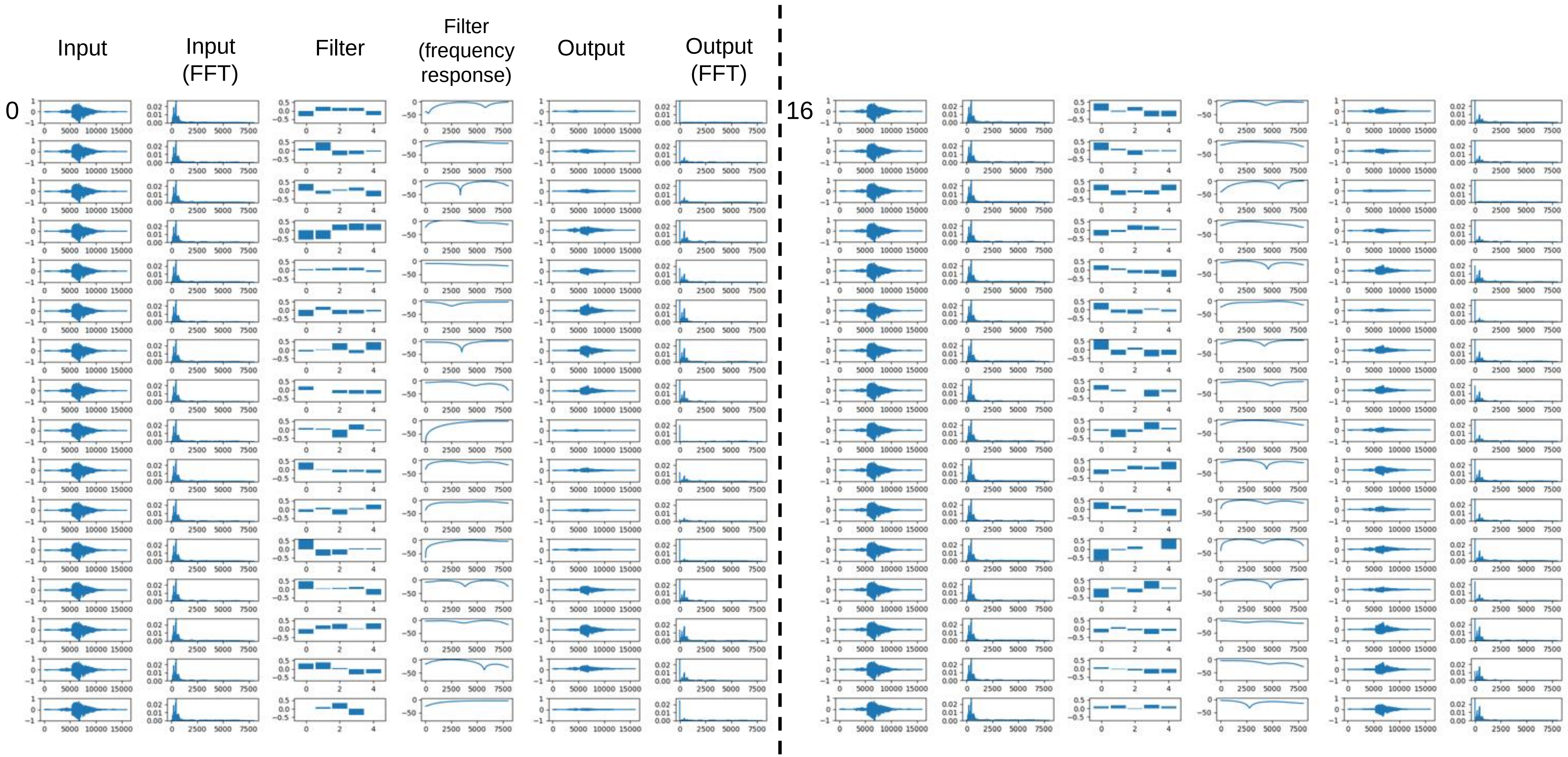
Audio Classification

- The most of Signal Processing analysis assume that Linear Time Invariant(LTI) System.
- Because of 'Relu' function is non-linear function, So, CNN is also non-linear system.
- But, how about before the first Relu?
- I think we can apply the Signal Processing technique before the first Relu layer.

Layer (type)	Output Shape	Param #
conv1d_67 (Conv1D)	(None, 16000, 32)	192
activation_67 (Activation)	(None, 16000, 32)	0
conv1d_68 (Conv1D)	(None, 16000, 32)	5152
activation_68 (Activation)	(None, 16000, 32)	0
max_pooling1d_55 (MaxPooling	(None, 5333, 32)	0
conv1d_69 (Conv1D)	(None, 5333, 32)	5152
activation_69 (Activation)	(None, 5333, 32)	0
max_pooling1d_56 (MaxPooling	(None, 1777, 32)	0
conv1d_70 (Conv1D)	(None, 1777, 32)	5152
activation_70 (Activation)	(None, 1777, 32)	0
max_pooling1d_57 (MaxPooling	(None, 592, 32)	0
conv1d_71 (Conv1D)	(None, 592, 64)	10304
activation_71 (Activation)	(None, 592, 64)	0
max_pooling1d_58 (MaxPooling	(None, 197, 64)	0
conv1d_72 (Conv1D)	(None, 197, 64)	20544
activation_72 (Activation)	(None, 197, 64)	0
max_pooling1d_59 (MaxPooling	(None, 65, 64)	0
conv1d_73 (Conv1D)	(None, 65, 64)	20544
activation_73 (Activation)	(None, 65, 64)	0
max_pooling1d_60 (MaxPooling	(None, 21, 64)	0
conv1d_74 (Conv1D)	(None, 21, 64)	20544
activation_74 (Activation)	(None, 21, 64)	0
max_pooling1d_61 (MaxPooling	(None, 7, 64)	0
flatten_12 (Flatten)	(None, 448)	0
dropout_12 (Dropout)	(None, 448)	0
dense_12 (Dense)	(None, 16)	7184

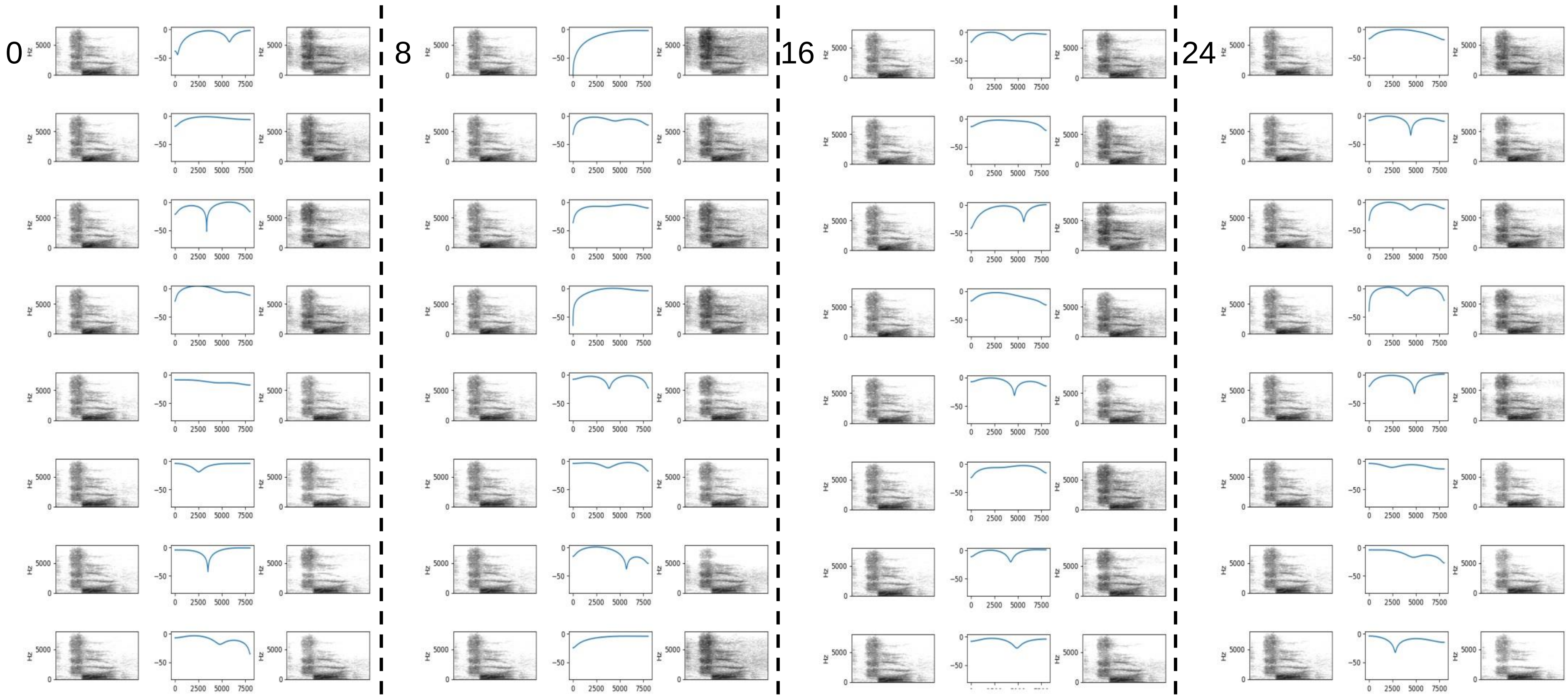
Audio Classification

- I want to know what does each filter to do.
- So I tried to analysis in frequency area.
- But, Fast Fourior Transform (FFT) is not good to analysis speech signal……



Audio Classification

- So I created spectrogram by Short Time Fourier Transform (STFT).
- Window Size: 512, Stride: 128
- We can see that it is applied to the shape of frequency response.



Any Question?

Thank you