ESC实验进展记录

1.调研论文

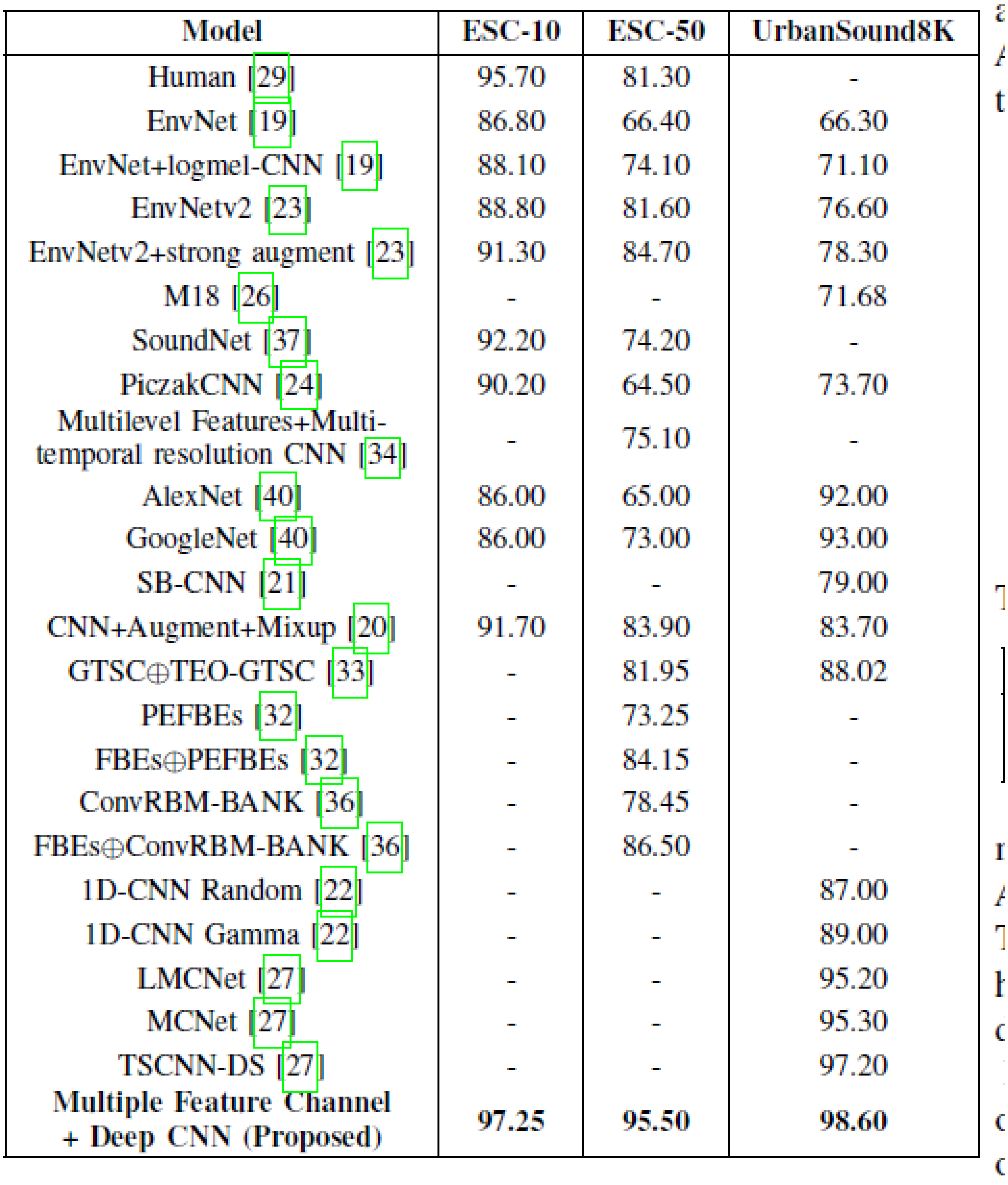
EnvNet

```
https://www.mi.t.u-
tokyo.ac.jp/assets/publication/LEARNING_ENVIRONMENTAL_SOUNDS_WITH_END-TO-
END_CONVOLUTIONAL_NEURAL_NETWORK.pd
```

EnvNet2

```
https://openreview.net/pdf?id=B1Gi6LeRZ
```

各模型实验结果



Model	ESC-10	ESC-50	UrbanSound8K
Human [29]	95.70	81.30	-
EnvNet [19]	86.80	66.40	66.30
EnvNet+logmel-CNN [19]	88.10	74.10	71.10
EnvNetv2 [23]	88.80	81.60	76.60
EnvNetv2+strong augment [23]	91.30	84.70	78.30
M18 [26]	-	-	71.68
SoundNet [37]	92.20	74.20	-
PiczakCNN [24]	90.20	64.50	73.70
Multilevel Features+Multi-temporal resolution CNN [34]	-	75.10	-
AlexNet [40]	86.00	65.00	92.00
GoogleNet [40]	86.00	73.00	93.00
SB-CNN [21]	-	-	79.00
CNN+Augment+Mixup [20]	91.70	83.90	83.70
GTSC⊕TEO-GTSC [33]	-	81.95	88.02
PEFBEs [32]	-	73.25	-
FBEs⊕PEFBEs [32]	-	84.15	-
ConvRBM-BANK [36]	-	78.45	-
FBEs⊕ConvRBM-BANK [36]	-	86.50	-
1D-CNN Random [22]	-	-	87.00
1D-CNN Gamma [22]	-	-	89.00
LMCNet [27]	-	-	95.20
MCNet [27]	-	-	95.30
TSCNN-DS [27]	-	-	97.20
Multiple Feature Channel + Deep CNN (Proposed)	**97.25**	**95.50**	**98.60**

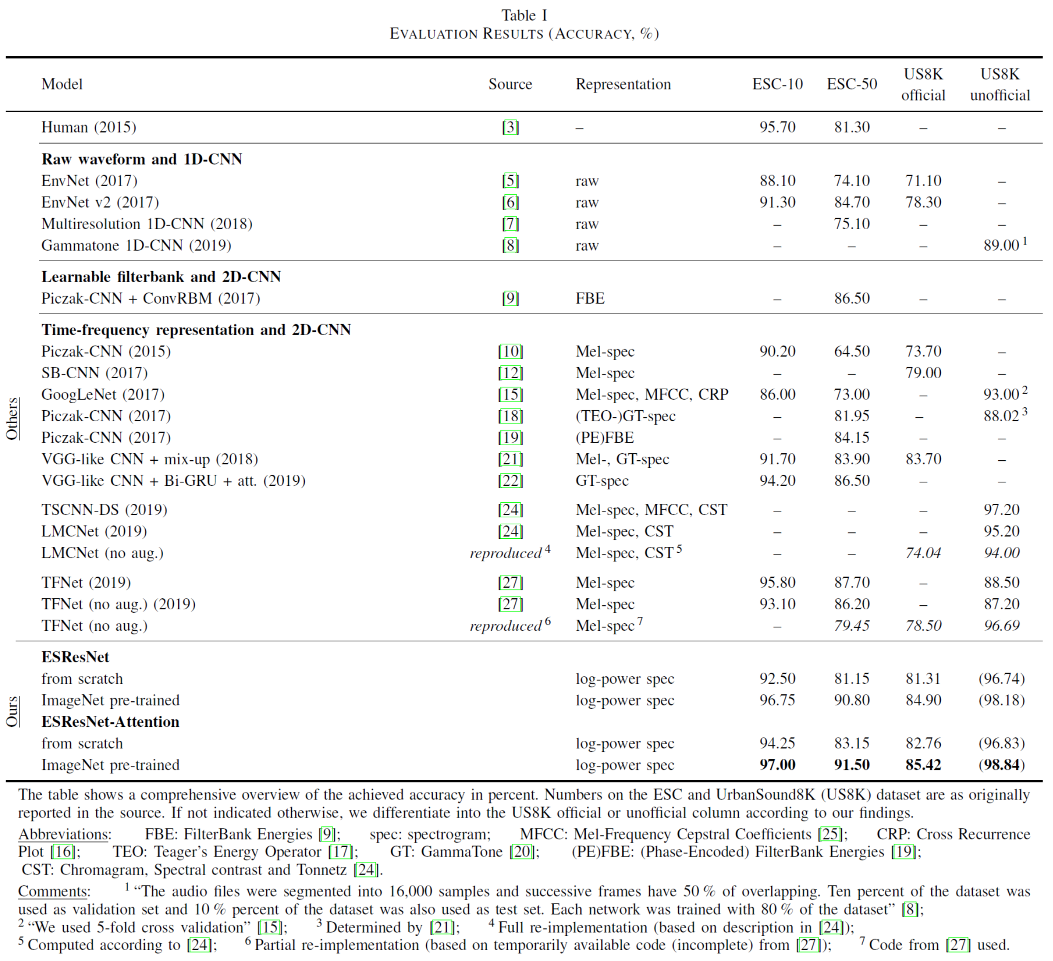


Table I
EVALUATION RESULTS (ACCURACY, %)

	Model	Source	Representation	ESC-10	ESC-50	US8K official	US8K unofficial
Others	Human (2015)	[3]	–	95.70	81.30	–	–
	Raw waveform and 1D-CNN						
	EnvNet (2017)	[5]	raw	88.10	74.10	71.10	–
	EnvNet v2 (2017)	[6]	raw	91.30	84.70	78.30	–
	Multiresolution 1D-CNN (2018)	[7]	raw	–	75.10	–	–
	Gammatone 1D-CNN (2019)	[8]	raw	–	–	–	89.00 [1]
	Learnable filterbank and 2D-CNN						
	Piczak-CNN + ConvRBM (2017)	[9]	FBE	–	86.50	–	–
	Time-frequency representation and 2D-CNN						
	Piczak-CNN (2015)	[10]	Mel-spec	90.20	64.50	73.70	–
	SB-CNN (2017)	[12]	Mel-spec	–	–	79.00	–
	GoogLeNet (2017)	[15]	Mel-spec, MFCC, CRP	86.00	73.00	–	93.00 [2]
	Piczak-CNN (2017)	[18]	(TEO-)GT-spec	–	81.95	–	88.02 [3]
	Piczak-CNN (2017)	[19]	(PE)FBE	–	84.15	–	–
	VGG-like CNN + mix-up (2018)	[21]	Mel-, GT-spec	91.70	83.90	83.70	–
	VGG-like CNN + Bi-GRU + att. (2019)	[22]	GT-spec	94.20	86.50	–	–
	TSCNN-DS (2019)	[24]	Mel-spec, MFCC, CST	–	–	–	97.20
	LMCNet (2019)	[24]	Mel-spec, CST	–	–	–	95.20
	LMCNet (no aug.)	*reproduced* [4]	Mel-spec, CST [5]	–	–	*74.04*	*94.00*
	TFNet (2019)	[27]	Mel-spec	95.80	87.70	–	88.50
	TFNet (no aug.) (2019)	[27]	Mel-spec	93.10	86.20	–	87.20
	TFNet (no aug.)	*reproduced* [6]	Mel-spec [7]	–	*79.45*	*78.50*	*96.69*
Ours	**ESResNet**						
	from scratch		log-power spec	92.50	81.15	81.31	(96.74)
	ImageNet pre-trained		log-power spec	96.75	90.80	84.90	(98.18)
	ESResNet-Attention						
	from scratch		log-power spec	94.25	83.15	82.76	(96.83)
	ImageNet pre-trained		log-power spec	**97.00**	**91.50**	**85.42**	(**98.84**)

The table shows a comprehensive overview of the achieved accuracy in percent. Numbers on the ESC and UrbanSound8K (US8K) dataset are as originally reported in the source. If not indicated otherwise, we differentiate into the US8K official or unofficial column according to our findings.

Abbreviations: FBE: FilterBank Energies [9]; spec: spectrogram; MFCC: Mel-Frequency Cepstral Coefficients [25]; CRP: Cross Recurrence Plot [16]; TEO: Teager's Energy Operator [17]; GT: GammaTone [20]; (PE)FBE: (Phase-Encoded) FilterBank Energies [19]; CST: Chromagram, Spectral contrast and Tonnetz [24].

Comments: [1] "The audio files were segmented into 16,000 samples and successive frames have 50 % of overlapping. Ten percent of the dataset was used as validation set and 10 % percent of the dataset was also used as test set. Each network was trained with 80 % of the dataset" [8]; [2] "We used 5-fold cross validation" [15]; [3] Determined by [21]; [4] Full re-implementation (based on description in [24]); [5] Computed according to [24]; [6] Partial re-implementation (based on temporarily available code (incomplete) from [27]); [7] Code from [27] used.

2.复现EnvNet2

EnvNet2 网络结构

```
class EnvNet2(nn.Module):
    def __init__(self, n_classes):
        super(EnvNet2, self).__init__()
        self.model = nn.Sequential(OrderedDict([
            ('conv1', EnvReLu(in_channels=1,
                              out_channels=32,
                              kernel_size=(1, 64),
                              stride=(1, 2),
                              padding=0)), #[b,32, 1, 33294]
            ('conv2', EnvReLu(in_channels=32,
                              out_channels=64,
                              kernel_size=(1, 16),
                              stride=(1, 2),
                              padding=0)), #[b, 64, 1, 16640]
            ('max_pool2', nn.MaxPool2d(kernel_size=(1, 64),
                                       stride=(1, 64),
                                       ceil_mode=True)), #[b, 64, 1, 260]
            ('transpose', Transpose()), #[b, 1, 64, 260]
            ('conv3', EnvReLu(in_channels=1,
                              out_channels=32,
                              kernel_size=(8, 8),
                              stride=(1, 1),
                              padding=0)), # [b, 32, 57, 253]
            ('conv4', EnvReLu(in_channels=32,
                              out_channels=32,
                              kernel_size=(8, 8),
                              stride=(1, 1),
                              padding=0)), #[b, 32, 50, 246]
            ('max_pool4', nn.MaxPool2d(kernel_size=(5, 3),
                                       stride=(5, 3),
                                       ceil_mode=True)), #[b, 32, 10, 82]
            ('conv5', EnvReLu(in_channels=32,
                              out_channels=64,
                              kernel_size=(1, 4),
                              stride=(1, 1),
                              padding=0)), #[b, 64, 10, 79]
            ('conv6', EnvReLu(in_channels=64,
                              out_channels=64,
                              kernel_size=(1, 4),
                              stride=(1, 1),
                              padding=0)),#[b, 64, 10, 76]
            ('max_pool6', nn.MaxPool2d(kernel_size=(1, 2),
                                       stride=(1, 2),
                                       ceil_mode=True)), #[b, 64, 10, 38]
            ('conv7', EnvReLu(in_channels=64,
                              out_channels=128,
                              kernel_size=(1, 2),
                              stride=(1, 1),
                              padding=0)), #[b, 128, 10, 37]
            ('conv8', EnvReLu(in_channels=128,
                              out_channels=128,
                              kernel_size=(1, 2),
                              stride=(1, 1),
                              padding=0)),#[b, 128, 10, 36]
            ('max_pool8', nn.MaxPool2d(kernel_size=(1, 2),
                                       stride=(1, 2),
                                       ceil_mode=True)), #[b, 128, 10, 18]
            ('conv9', EnvReLu(in_channels=128,
                              out_channels=256,
                              kernel_size=(1, 2),
                              stride=(1, 1),
                              padding=0)),#[b, 256, 10, 17]
            ('conv10', EnvReLu(in_channels=256,
                               out_channels=256,
                               kernel_size=(1, 2),
                               stride=(1, 1),
                               padding=0)),#[b, 256, 10, 16]
            ('max_pool10', nn.MaxPool2d(kernel_size=(1, 2),
                                        stride=(1, 2),
                                        ceil_mode=True)), #[b, 256, 10, 8]
            ('flatten', Flatten()),
            ('fc11', nn.Linear(in_features=256 * 10 * 8, out_features=4096,
bias=True)), #[b, 20480]
            ('relu11', nn.ReLU()),
            ('dropout11', nn.Dropout()),
            ('fc12', nn.Linear(in_features=4096, out_features=4096, bias=True)),
#[2, 4096]
            ('relu12', nn.ReLU()),
            ('dropout12', nn.Dropout()),
            ('fc13', nn.Linear(in_features=4096, out_features=n_classes,
bias=True))#[2, 50]
        ]))
```

```
    def forward(self, x):
        #x [b, 1, ,1, 66650]

        return self.model(x)
```

在 ESC10数据集上结果

```
+------------------------------+
| Sound classification
+------------------------------+
| dataset  : esc10
| netType  : EnvNet2
| learning : BC
| augment  : True
| nEpochs  : 1200
| LRInit   : 0.01
| batchSize: 64
| optimizer  : SGD
| nesterov   : True
| milestones : [600, 900]
| gamma : 0.1
+------------------------------+
Data Get fold 1
Epoch [1196/1200], train_loss: 0.4655, val_loss: 0.4022, Accuracy: 86.2500
Epoch [1197/1200], train_loss: 0.4163, val_loss: 0.3937, Accuracy: 88.7500
Epoch [1198/1200], train_loss: 0.4209, val_loss: 0.4021, Accuracy: 86.2500
Epoch [1199/1200], train_loss: 0.4428, val_loss: 0.4042, Accuracy: 87.5000
Epoch [1200/1200], train_loss: 0.4651, val_loss: 0.4087, Accuracy: 86.2500
+------------------------------+
Data Get fold 2
Epoch [1196/1200], train_loss: 0.3993, val_loss: 0.3047, Accuracy: 97.5000
Epoch [1197/1200], train_loss: 0.3958, val_loss: 0.3071, Accuracy: 97.5000
Epoch [1198/1200], train_loss: 0.3883, val_loss: 0.3081, Accuracy: 97.5000
Epoch [1199/1200], train_loss: 0.4329, val_loss: 0.3049, Accuracy: 97.5000
Epoch [1200/1200], train_loss: 0.4068, val_loss: 0.3053, Accuracy: 97.5000
+------------------------------+
Data Get fold 3
Epoch [1196/1200], train_loss: 0.3970, val_loss: 0.4347, Accuracy: 92.5000
Epoch [1197/1200], train_loss: 0.4031, val_loss: 0.4404, Accuracy: 92.5000
Epoch [1198/1200], train_loss: 0.4106, val_loss: 0.4339, Accuracy: 92.5000
Epoch [1199/1200], train_loss: 0.4361, val_loss: 0.4307, Accuracy: 92.5000
Epoch [1200/1200], train_loss: 0.4797, val_loss: 0.4386, Accuracy: 92.5000
+------------------------------+
Data Get fold 4
Epoch [1196/1200], train_loss: 0.4108, val_loss: 0.3098, Accuracy: 97.5000
Epoch [1197/1200], train_loss: 0.4198, val_loss: 0.3087, Accuracy: 97.5000
Epoch [1198/1200], train_loss: 0.4164, val_loss: 0.3116, Accuracy: 97.5000
Epoch [1199/1200], train_loss: 0.3827, val_loss: 0.3127, Accuracy: 97.5000
Epoch [1200/1200], train_loss: 0.4021, val_loss: 0.3126, Accuracy: 97.5000
+------------------------------+
Data Get fold 5
Epoch [1196/1200], train_loss: 0.3586, val_loss: 0.3725, Accuracy: 93.7500
Epoch [1197/1200], train_loss: 0.3973, val_loss: 0.3718, Accuracy: 93.7500
Epoch [1198/1200], train_loss: 0.4139, val_loss: 0.3700, Accuracy: 93.7500
Epoch [1199/1200], train_loss: 0.3955, val_loss: 0.3774, Accuracy: 93.7500
Epoch [1200/1200], train_loss: 0.3722, val_loss: 0.3780, Accuracy: 93.7500
```

在ESC50数据集上结果

Max Accuracy = 80.75 Max Accuracy = 83.75 Max Accuracy = 84.75 Max Accuracy = 89.00 Max Accuracy = 83.75

```
+------------------------------+
| Sound classification
+------------------------------+
| dataset  : esc50
| netType  : EnvNet2
| learning : BC
| augment  : True
| nEpochs  : 1600
| LRInit   : 0.1
| batchSize: 64
| optimizer  : SGD
| nesterov   : True
| milestones : [480, 960, 1440]
| gamma : 0.1
+------------------------------+
Data Get fold 1
Elapsed [1 day, 3:38:17], Epoch [1596/1600], Train_loss = 1.0274, Val_loss =
0.9388, Accuracy = 79.5000
Elapsed [1 day, 3:39:19], Epoch [1597/1600], Train_loss = 1.0193, Val_loss =
0.9283, Accuracy = 79.2500
Elapsed [1 day, 3:40:21], Epoch [1598/1600], Train_loss = 1.0118, Val_loss =
0.9333, Accuracy = 79.2500
Elapsed [1 day, 3:41:24], Epoch [1599/1600], Train_loss = 1.0436, Val_loss =
0.9363, Accuracy = 79.0000
Elapsed [1 day, 3:42:26], Epoch [1600/1600], Train_loss = 1.0241, Val_loss =
0.9354, Accuracy = 79.2500
+------------------------------+
Data Get fold 2
Elapsed [1 day, 3:31:05], Epoch [1596/1600], Train_loss = 0.9443, Val_loss =
0.8874, Accuracy = 81.2500
Elapsed [1 day, 3:32:07], Epoch [1597/1600], Train_loss = 0.9494, Val_loss =
0.8878, Accuracy = 82.7500
Elapsed [1 day, 3:33:09], Epoch [1598/1600], Train_loss = 0.9792, Val_loss =
0.8850, Accuracy = 82.7500
Elapsed [1 day, 3:34:11], Epoch [1599/1600], Train_loss = 0.9706, Val_loss =
0.8882, Accuracy = 81.5000
Elapsed [1 day, 3:35:13], Epoch [1600/1600], Train_loss = 0.9647, Val_loss =
0.8931, Accuracy = 82.0000
+------------------------------+
Data Get fold 3
Elapsed [1 day, 3:22:10], Epoch [1596/1600], Train_loss = 1.0307, Val_loss =
0.8564, Accuracy = 82.7500
Elapsed [1 day, 3:23:12], Epoch [1597/1600], Train_loss = 1.0204, Val_loss =
0.8605, Accuracy = 82.5000
Elapsed [1 day, 3:24:13], Epoch [1598/1600], Train_loss = 1.0164, Val_loss =
0.8584, Accuracy = 82.2500
Elapsed [1 day, 3:25:14], Epoch [1599/1600], Train_loss = 0.9903, Val_loss =
0.8648, Accuracy = 82.7500
Elapsed [1 day, 3:26:16], Epoch [1600/1600], Train_loss = 1.0001, Val_loss =
0.8605, Accuracy = 82.5000
+------------------------------+
Data Get fold 4
```

```
Elapsed [1 day, 3:23:06], Epoch [1596/1600], Train_loss = 1.0214, Val_loss =
0.7432, Accuracy = 88.0000
Elapsed [1 day, 3:24:08], Epoch [1597/1600], Train_loss = 1.0337, Val_loss =
0.7420, Accuracy = 87.7500
Elapsed [1 day, 3:25:10], Epoch [1598/1600], Train_loss = 1.0312, Val_loss =
0.7443, Accuracy = 88.7500
Elapsed [1 day, 3:26:12], Epoch [1599/1600], Train_loss = 1.0470, Val_loss =
0.7468, Accuracy = 88.7500
Elapsed [1 day, 3:27:14], Epoch [1600/1600], Train_loss = 1.0000, Val_loss =
0.7415, Accuracy = 88.2500
+-----------------------------+
Data Get fold 5
Elapsed [1 day, 0:53:54], Epoch [1596/1600], Train_loss = 1.0361, Val_loss =
0.8867, Accuracy = 82.5000
Elapsed [1 day, 0:54:49], Epoch [1597/1600], Train_loss = 1.0463, Val_loss =
0.8850, Accuracy = 81.7500
Elapsed [1 day, 0:55:45], Epoch [1598/1600], Train_loss = 1.0815, Val_loss =
0.8843, Accuracy = 82.5000
Elapsed [1 day, 0:56:40], Epoch [1599/1600], Train_loss = 1.0368, Val_loss =
0.8861, Accuracy = 82.5000
Elapsed [1 day, 0:57:36], Epoch [1600/1600], Train_loss = 1.0375, Val_loss =
0.8875, Accuracy = 82.5000
```

3.调研EnvNet2的鲁棒性

1. 改变音量

   把 ESC50 fold 1 得到的模型 验证所有的音频音量全部+10dB或者-10dB，验证准确率

   准确率下降 1.5% 左右

2.加背景噪音

把 ESC50 fold 1 得到的模型 验证 所有的音频加入适当背景音，例如说话声，音乐声音，或白噪声

1.加 绝对分贝的噪音 45dB 50dB 55dB 的说话声、音乐声、白噪音

2.加 相对分贝噪音 -10dB 的说话声、音乐声、白噪音

准确率下降 10% 左右

4.EnvNet3

DenseNet **EnvNet2 + DenseNet**

```
https://arxiv.org/pdf/1608.06993.pdf
```

```
Envnet3_1   DenseNet(growth_rate=16, block_config=(3, 6, 12, 8)))

Envnet3_2   DenseNet(growth_rate=32, block_config=(6, 12, 24, 16),
drop_rate=0.5) + nn.AdaptiveAvgPool2d(1)

Envnet3_3   DenseNet(growth_rate=16, block_config=(6, 12, 24, 16),
drop_rate=0.5)) + nn.AdaptiveAvgPool2d(1)

Envnet3_4   DenseNet(growth_rate=16, block_config=(6, 12, 24, 16))) +
nn.AdaptiveAvgPool2d(1)

Envnet3_5   DenseNet(growth_rate=32, block_config=(6, 12, 24, 16))) +
nn.AdaptiveAvgPool2d(1)
```

EnvNet3 网络结构

```
class EnvNet3(nn.Module):
    def __init__(self, n_classes):
        super(EnvNet3, self).__init__()
        self.model = nn.Sequential(OrderedDict([
            ('conv1', EnvReLu(in_channels=1,
                              out_channels=32,
                              kernel_size=(1, 64),
                              stride=(1, 2),
                              padding=0)), #[b,32, 1, 33294]
            ('conv2', EnvReLu(in_channels=32,
                              out_channels=64,
                              kernel_size=(1, 16),
                              stride=(1, 2),
                              padding=0)), #[b, 64, 1, 16640]
            ('max_pool2', nn.MaxPool2d(kernel_size=(1, 64),
                                       stride=(1, 64),
                                       ceil_mode=True)), #[b, 64, 1, 260]
            ('transpose', Transpose()), #[b, 1, 64, 260]

            ('densenet', DenseNet(growth_rate=16, block_config=(3, 6, 12, 8))),

            ('flatten', Flatten()),
            ('fc11', nn.Linear(in_features=262 * 4 * 32, out_features=1024,
bias=True)),
            ('relu11', nn.ReLU()),
            ('dropout11', nn.Dropout()),
            ('fc12', nn.Linear(in_features=1024, out_features=n_classes,
bias=True))#[b, 50]
            ]))

    def forward(self, x):

        return self.model(x)
```

Envnet3_1

ESC50 数据集 fold 1 结果 **Max Accuracy = 82.00**

```
+------------------------------+
| Sound classification
```

```
+------------------------------+
| dataset  : esc50
| netType  : EnvNet3_1
| learning : BC
| augment  : True
| nEpochs  : 2000
| LRInit   : 0.001
| batchSize: 16
| optimizer   : Adam
| beta1   : 0.9
| beta2 : 0.999
| eps : 1e-08
| amsgrad : True
+------------------------------+
Data Get fold 1
Elapsed [1 day, 21:13:25], Epoch [1996/2000], Train_loss = 0.8722, Val_loss = 0.9259, Accuracy = 80.2500
Elapsed [1 day, 21:14:47], Epoch [1997/2000], Train_loss = 0.8640, Val_loss = 0.9565, Accuracy = 77.7500
Elapsed [1 day, 21:16:08], Epoch [1998/2000], Train_loss = 0.9338, Val_loss = 0.9378, Accuracy = 77.2500
Elapsed [1 day, 21:17:28], Epoch [1999/2000], Train_loss = 0.8848, Val_loss = 0.9182, Accuracy = 79.0000
Elapsed [1 day, 21:18:51], Epoch [2000/2000], Train_loss = 0.9255, Val_loss = 0.9554, Accuracy = 78.5000
```

EnvNet3_2网络结构

```
class EnvNet3_2(nn.Module):
    def __init__(self, n_classes):
        super(EnvNet3.1, self).__init__()
        self.model = nn.Sequential(OrderedDict([
            ('conv1', EnvReLu(in_channels=1,
                              out_channels=32,
                              kernel_size=(1, 64),
                              stride=(1, 2),
                              padding=0)), #[b,32, 1, 33294]
            ('conv2', EnvReLu(in_channels=32,
                              out_channels=64,
                              kernel_size=(1, 16),
                              stride=(1, 2),
                              padding=0)), #[b, 64, 1, 16640]
            ('max_pool2', nn.MaxPool2d(kernel_size=(1, 64),
                                       stride=(1, 64),
                                       ceil_mode=True)), #[b, 64, 1, 260]
            ('transpose', Transpose()), #[b, 1, 64, 260]


            ('densenet', DenseNet(growth_rate=32, block_config=(6, 12, 24, 16), drop_rate=0.5)), #[b, 1024,4,16]
            # growth_rate= 32 --> 16 解决当前服务器内存不够

            ('global avepool', nn.AdaptiveAvgPool2d(1)),#[b, 1024,1,1]

            ('flatten', Flatten()),
            ('fc13', nn.Linear(in_features=1024, out_features=n_classes, bias=True))#[b, 50]
            ]))

    def forward(self, x):

        return self.model(x)
```

EnvNet3_2

ESC50 数据集 fold 1 结果 **Max Accuracy = 78.75**

```
+------------------------------+
| Sound classification
+------------------------------+
 dataset: esc50
 netType: EnvNet3_2
 learning: BC
 augment: True
 nEpochs: 2000
 LRInit: 0.001
 batchSize: 16
 optimizer: Adam
  beta1: 0.9
 beta2: 0.999
 eps: 1e-08
 amsgrad: True
+------------------------------+
Data Get fold 1
Elapsed [1 day, 23:12:23], Epoch [1996/2000], Train_loss = 0.6282, Val_loss = 1.1457, Accuracy = 69.5000
Elapsed [1 day, 23:13:47], Epoch [1997/2000], Train_loss = 0.6066, Val_loss = 1.0071, Accuracy = 76.2500
Elapsed [1 day, 23:15:12], Epoch [1998/2000], Train_loss = 0.6121, Val_loss = 0.9933, Accuracy = 76.0000
Elapsed [1 day, 23:16:40], Epoch [1999/2000], Train_loss = 0.6178, Val_loss = 0.9951, Accuracy = 76.2500
Elapsed [1 day, 23:18:05], Epoch [2000/2000], Train_loss = 0.6194, Val_loss = 0.9856, Accuracy = 74.0000
```

EnvNet3_3

ESC50 数据集 fold 1 结果 **Max Accuracy = 77.50**

```
+------------------------------+
| Sound classification
+------------------------------+
 dataset: esc50
 netType: EnvNet3_3
 learning: BC
 augment: True
 nEpochs: 1600
 LRInit: 0.001
 batchSize: 16
 optimizer: Adam
  beta1: 0.9
 beta2: 0.999
 eps: 1e-08
 amsgrad: True
+------------------------------+
Data Get fold 1
Elapsed [2 days, 2:06:03], Epoch [1596/1600], Train_loss = 0.8138, Val_loss =
1.1583, Accuracy = 74.0000
Elapsed [2 days, 2:07:58], Epoch [1597/1600], Train_loss = 0.8148, Val_loss =
1.4899, Accuracy = 64.0000
Elapsed [2 days, 2:09:49], Epoch [1598/1600], Train_loss = 0.8226, Val_loss =
1.2007, Accuracy = 72.2500
Elapsed [2 days, 2:11:43], Epoch [1599/1600], Train_loss = 0.7951, Val_loss =
1.1426, Accuracy = 70.7500
Elapsed [2 days, 2:13:34], Epoch [1600/1600], Train_loss = 0.8318, Val_loss =
1.4835, Accuracy = 62.0000
```

EnvNet3_4

ESC50 数据集 fold 1 结果

Max Accuracy = 86.75 Max Accuracy = 87.00 Max Accuracy = 87.50 Max Accuracy = 89.75 Max Accuracy = 88.25

平均准确率：**Mean Accuracy = 87.85**

```
+------------------------------+
| Sound classification
+------------------------------+
 dataset: esc50
 netType: EnvNet3_4
 learning: BC
 augment: True
 nEpochs: 1600
 LRInit: 0.001
 batchSize: 16
 optimizer: Adam
 beta1: 0.9
 beta2: 0.999
 eps: 1e-08
 amsgrad: True
 milestones: [960, 1280]
 gamma: 0.1
+------------------------------+
Data Get fold 1
Elapsed [2 days, 2:02:40], Epoch [1596/1600], Train_loss = 0.6177, Val_loss =
0.7840, Accuracy = 84.7500
Elapsed [2 days, 2:04:35], Epoch [1597/1600], Train_loss = 0.6368, Val_loss =
0.8022, Accuracy = 85.0000
Elapsed [2 days, 2:06:26], Epoch [1598/1600], Train_loss = 0.6215, Val_loss =
0.8209, Accuracy = 85.2500
Elapsed [2 days, 2:08:18], Epoch [1599/1600], Train_loss = 0.5985, Val_loss =
0.8001, Accuracy = 85.2500
Elapsed [2 days, 2:10:09], Epoch [1600/1600], Train_loss = 0.6124, Val_loss =
0.7852, Accuracy = 83.7500
+------------------------------+
Data Get fold 2
Elapsed [1 day, 8:27:00], Epoch [1196/1200], Train_loss = 0.6410, Val_loss =
0.7502, Accuracy = 84.7500
```

```
Elapsed [1 day, 8:28:38], Epoch [1197/1200], Train_loss = 0.6562, Val_loss =
0.7404, Accuracy = 84.7500
Elapsed [1 day, 8:30:15], Epoch [1198/1200], Train_loss = 0.6657, Val_loss =
0.7402, Accuracy = 85.5000
Elapsed [1 day, 8:31:53], Epoch [1199/1200], Train_loss = 0.6819, Val_loss =
0.7256, Accuracy = 85.5000
Elapsed [1 day, 8:33:31], Epoch [1200/1200], Train_loss = 0.6642, Val_loss =
0.7536, Accuracy = 84.0000
+------------------------------+
Data Get fold 3
Elapsed [1 day, 8:46:49], Epoch [1196/1200], Train_loss = 0.6764, Val_loss =
0.7311, Accuracy = 85.5000
Elapsed [1 day, 8:48:28], Epoch [1197/1200], Train_loss = 0.6690, Val_loss =
0.7290, Accuracy = 86.2500
Elapsed [1 day, 8:50:06], Epoch [1198/1200], Train_loss = 0.6627, Val_loss =
0.7502, Accuracy = 84.7500
Elapsed [1 day, 8:51:45], Epoch [1199/1200], Train_loss = 0.6811, Val_loss =
0.7386, Accuracy = 86.0000
Elapsed [1 day, 8:53:23], Epoch [1200/1200], Train_loss = 0.6631, Val_loss =
0.7510, Accuracy = 84.2500
+------------------------------+
Data Get fold 4
Elapsed [1 day, 12:11:26], Epoch [1196/1200], Train_loss = 0.7004, Val_loss =
0.5961, Accuracy = 88.7500
Elapsed [1 day, 12:13:18], Epoch [1197/1200], Train_loss = 0.6596, Val_loss =
0.5900, Accuracy = 87.5000
Elapsed [1 day, 12:15:10], Epoch [1198/1200], Train_loss = 0.6669, Val_loss =
0.6117, Accuracy = 87.5000
Elapsed [1 day, 12:17:01], Epoch [1199/1200], Train_loss = 0.6610, Val_loss =
0.6163, Accuracy = 87.5000
Elapsed [1 day, 12:18:53], Epoch [1200/1200], Train_loss = 0.6758, Val_loss =
0.6012, Accuracy = 88.7500
+------------------------------+
Data Get fold 5
Elapsed [1 day, 12:59:47], Epoch [1196/1200], Train_loss = 0.6038, Val_loss =
0.7174, Accuracy = 87.0000
Elapsed [1 day, 13:01:38], Epoch [1197/1200], Train_loss = 0.6199, Val_loss =
0.7226, Accuracy = 86.5000
Elapsed [1 day, 13:03:29], Epoch [1198/1200], Train_loss = 0.6245, Val_loss =
0.7019, Accuracy = 87.0000
Elapsed [1 day, 13:05:19], Epoch [1199/1200], Train_loss = 0.6412, Val_loss =
0.7133, Accuracy = 87.5000
Elapsed [1 day, 13:07:10], Epoch [1200/1200], Train_loss = 0.6044, Val_loss =
0.7290, Accuracy = 86.0000
```

5.EnvNet4

Sample-Level CNN Architectures for Music Auto-Tagging Using Raw Waveforms

加入 SENet + Multi-Scale + Densenet + Global Avg Pooling

```
https://arxiv.org/pdf/1710.10451.pdf
```

```
Envnet4_1   DenseNet(growth_rate=32, block_config=(3, 6, 12, 8))) + SEBlock +
MultiScale + nn.AdaptiveAvgPool2d(1)

Envnet4_2   DenseNet(growth_rate=32, block_config=(3, 6, 12)) + SEBlock +
MultiScale + nn.AdaptiveAvgPool2d(1)

Envnet4_3   DenseNet(growth_rate=32, block_config=(3, 6, 12), drop_rate=0.5) +
+ SEBlock + MultiScale + nn.AdaptiveAvgPool2d(1)

Envnet4_4   DenseNet(growth_rate=16, block_config=(6, 12, 24, 16))) + SELayer +
MultiScale + nn.AdaptiveAvgPool2d(1)

Envnet4_5   DenseNet(growth_rate=16, block_config=(6, 12, 24, 16))) + SEBlock +
MultiScale + nn.AdaptiveAvgPool2d(1)
```

EnvNet4 网络结构

```
class EnvNet4(nn.Module):
    def __init__(self, n_classes):
        super(EnvNet4, self).__init__()
        self.model = nn.Sequential(OrderedDict([
            ('conv1', EnvReLu(in_channels=1,
                              out_channels=32,
                              kernel_size=(1, 64),
                              stride=(1, 2),
                              padding=0)), #[b,32, 1, 33294]
            ('conv2', EnvReLu(in_channels=32,
                              out_channels=64,
                              kernel_size=(1, 16),
                              stride=(1, 2),
                              padding=0)), #[b, 64, 1, 16640]
            ('max_pool2', nn.MaxPool2d(kernel_size=(1, 64),
                                       stride=(1, 64),
                                       ceil_mode=True)), #[b, 64, 1, 260]
            ('transpose', Transpose()), #[b, 1, 64, 260]

            ('densenet', DenseNet(growth_rate=32, block_config=(3, 6, 12, 8))),
#[b, 912,8,32]

            ('senet1', SEBottleneck(inplanes=136, planes=136//4)), #[b,
136,8,32]
            ('senet2', SEBottleneck(inplanes=260, planes=260//4)),#[b, 260,8,32]
            ('global_max_pool2', nn.AdaptiveAvgPool2d(1)),#[b, *,1,1]

            ('flatten', Flatten()),
            ('fc13', nn.Linear(in_features=(136+260+516),
out_features=n_classes, bias=True))#[b,50]
            ]))

    def forward(self, x):
        for i in range(len(self.model)):
            if i < 4:
                x = self.model[i](x)
            if i == 4:
                out1, out2, out3 = self.model[i](x)
        out1 = self.model[8](out1)

        out2 = self.model[6](out2)
        out2 = self.model[8](out2)

        out3 = self.model[7](out3)
        out3 = self.model[8](out3)

        out = torch.cat((out1, out2, out3), dim=1)

        out = self.model[9](out)
        out = self.model[10](out)

        return out
```

EnvNet4_1

ESC50 数据集 fold 1 结果　**Max Accuracy = 85.75**

```
+------------------------------+
| Sound classification
+------------------------------+
 dataset: esc50
 netType: EnvNet4_1
 learning: BC
 augment: True
 nEpochs: 2000
 LRInit: 0.001
 batchSize: 16
 optimizer: Adam
  beta1: 0.9
 beta2: 0.999
 eps: 1e-08
 amsgrad: True
+------------------------------+
Data Get fold 1
Elapsed [2 days, 7:34:05], Epoch [1996/2000], Train_loss = 0.5840, Val_loss = 0.8498, Accuracy = 80.2500
Elapsed [2 days, 7:35:47], Epoch [1997/2000], Train_loss = 0.5696, Val_loss = 0.8305, Accuracy = 81.7500
Elapsed [2 days, 7:37:25], Epoch [1998/2000], Train_loss = 0.5603, Val_loss = 0.8307, Accuracy = 81.0000
Elapsed [2 days, 7:39:02], Epoch [1999/2000], Train_loss = 0.5752, Val_loss = 0.8081, Accuracy = 81.0000
Elapsed [2 days, 7:40:44], Epoch [2000/2000], Train_loss = 0.5710, Val_loss = 0.7889, Accuracy = 81.0000
```

EnvNet4_2

ESC50 数据集 fold 1 结果　**Max Accuracy = 85.00**

```
+------------------------------+
| Sound classification
+------------------------------+
 dataset: esc50
 netType: EnvNet4_2
 learning: BC
 augment: True
 nEpochs: 2000
 LRInit: 0.001
 batchSize: 32
 optimizer: Adam
  beta1: 0.9
 beta2: 0.999
 eps: 1e-08
 amsgrad: True
+------------------------------+
Data Get fold 1
Elapsed [1 day, 15:05:45], Epoch [1996/2000], Train_loss = 0.6204, Val_loss =
0.8237, Accuracy = 83.7500
Elapsed [1 day, 15:06:55], Epoch [1997/2000], Train_loss = 0.6006, Val_loss =
0.8762, Accuracy = 79.5000
Elapsed [1 day, 15:08:05], Epoch [1998/2000], Train_loss = 0.6229, Val_loss =
0.8795, Accuracy = 78.5000
Elapsed [1 day, 15:09:16], Epoch [1999/2000], Train_loss = 0.6166, Val_loss =
0.8913, Accuracy = 79.0000
Elapsed [1 day, 15:10:26], Epoch [2000/2000], Train_loss = 0.6080, Val_loss =
0.8719, Accuracy = 80.0000
```

EnvNet4_2

ESC50 数据集 结果

Max Accuracy = 86.50 Max Accuracy = 85.50 Max Accuracy = 85.75 Max Accuracy = 90.00 Max Accuracy = 86.00

```
+------------------------------+
| Sound classification
+------------------------------+
 dataset: esc50
 netType: EnvNet4_2
 learning: BC
 augment: True
 nEpochs: 2000
 LRInit: 0.001
 batchSize: 32
 optimizer: Adam
  beta1: 0.9
 beta2: 0.999
 eps: 1e-08
 amsgrad: True
+------------------------------+
Data Get fold 1
Elapsed [1 day, 13:59:55], Epoch [1596/1600], Train_loss = 0.6765, Val_loss =
0.8162, Accuracy = 81.5000
Elapsed [1 day, 14:01:12], Epoch [1597/1600], Train_loss = 0.6680, Val_loss =
0.7907, Accuracy = 83.5000
Elapsed [1 day, 14:02:29], Epoch [1598/1600], Train_loss = 0.6539, Val_loss =
0.7664, Accuracy = 84.2500
Elapsed [1 day, 14:03:45], Epoch [1599/1600], Train_loss = 0.6468, Val_loss =
0.7979, Accuracy = 82.0000
Elapsed [1 day, 14:05:01], Epoch [1600/1600], Train_loss = 0.6705, Val_loss =
0.9057, Accuracy = 77.7500
+------------------------------+
Data Get fold 2
Elapsed [1 day, 8:11:57], Epoch [1596/1600], Train_loss = 0.7140, Val_loss =
0.8086, Accuracy = 82.7500
Elapsed [1 day, 8:13:15], Epoch [1597/1600], Train_loss = 0.6767, Val_loss =
0.8025, Accuracy = 84.2500
Elapsed [1 day, 8:14:32], Epoch [1598/1600], Train_loss = 0.6790, Val_loss =
0.8313, Accuracy = 82.7500
Elapsed [1 day, 8:15:49], Epoch [1599/1600], Train_loss = 0.6555, Val_loss =
0.8396, Accuracy = 83.2500
Elapsed [1 day, 8:17:05], Epoch [1600/1600], Train_loss = 0.7019, Val_loss =
0.8921, Accuracy = 80.7500
+------------------------------+
Data Get fold 3
Elapsed [1 day, 3:37:57], Epoch [1596/1600], Train_loss = 0.7063, Val_loss =
0.7901, Accuracy = 83.7500
Elapsed [1 day, 3:38:59], Epoch [1597/1600], Train_loss = 0.7043, Val_loss =
0.8000, Accuracy = 83.0000
Elapsed [1 day, 3:40:01], Epoch [1598/1600], Train_loss = 0.6849, Val_loss =
0.8089, Accuracy = 83.2500
Elapsed [1 day, 3:41:04], Epoch [1599/1600], Train_loss = 0.6931, Val_loss =
0.8399, Accuracy = 82.0000
Elapsed [1 day, 3:42:06], Epoch [1600/1600], Train_loss = 0.6949, Val_loss =
0.8068, Accuracy = 83.5000
+------------------------------+
Data Get fold 4
Elapsed [1 day, 3:45:46], Epoch [1596/1600], Train_loss = 0.6974, Val_loss =
0.6410, Accuracy = 87.7500
Elapsed [1 day, 3:46:48], Epoch [1597/1600], Train_loss = 0.6673, Val_loss =
0.6970, Accuracy = 86.2500
Elapsed [1 day, 3:47:50], Epoch [1598/1600], Train_loss = 0.6725, Val_loss =
0.6439, Accuracy = 87.7500
Elapsed [1 day, 3:48:53], Epoch [1599/1600], Train_loss = 0.6866, Val_loss =
0.6489, Accuracy = 88.0000
Elapsed [1 day, 3:49:55], Epoch [1600/1600], Train_loss = 0.6705, Val_loss =
0.6487, Accuracy = 89.2500
+------------------------------+
Data Get fold 5
Elapsed [1 day, 5:40:05], Epoch [1596/1600], Train_loss = 0.6695, Val_loss =
0.8229, Accuracy = 78.2500
Elapsed [1 day, 5:41:06], Epoch [1597/1600], Train_loss = 0.6337, Val_loss =
0.8319, Accuracy = 80.5000
Elapsed [1 day, 5:42:08], Epoch [1598/1600], Train_loss = 0.6326, Val_loss =
0.7909, Accuracy = 82.7500
Elapsed [1 day, 5:43:10], Epoch [1599/1600], Train_loss = 0.6781, Val_loss =
0.7445, Accuracy = 85.2500
Elapsed [1 day, 5:44:11], Epoch [1600/1600], Train_loss = 0.6357, Val_loss =
0.7992, Accuracy = 81.5000
```

EnvNet4_3

ESC50 数据集 fold 1 结果 **Max Accuracy = 79.25**

```
+------------------------------+
| Sound classification
+------------------------------+
 dataset: esc50
 netType: EnvNet4_3
 learning: BC
 augment: True
 nEpochs: 2000
 LRInit: 0.001
 batchSize: 16
 optimizer: Adam
  beta1: 0.9
 beta2: 0.999
 eps: 1e-08
 amsgrad: True
+------------------------------+
Data Get fold 1
Elapsed [1 day, 17:40:49], Epoch [1996/2000], Train_loss = 0.8956, Val_loss = 1.1470, Accuracy = 71.7500
Elapsed [1 day, 17:42:08], Epoch [1997/2000], Train_loss = 0.8282, Val_loss = 1.1223, Accuracy = 71.5000
Elapsed [1 day, 17:43:22], Epoch [1998/2000], Train_loss = 0.8421, Val_loss = 1.0830, Accuracy = 74.0000
Elapsed [1 day, 17:44:36], Epoch [1999/2000], Train_loss = 0.8372, Val_loss = 1.0218, Accuracy = 75.5000
Elapsed [1 day, 17:45:55], Epoch [2000/2000], Train_loss = 0.8271, Val_loss = 1.1182, Accuracy = 73.5000
```

EnvNet4_4

ESC50 数据集 fold 1 结果 **Max Accuracy = 85.5**

```
+------------------------------+
| Sound classification
+------------------------------+
 dataset: esc50
 netType: EnvNet4_4
 learning: BC
 augment: True
 nEpochs: 1600
 LRInit: 0.001
 batchSize: 16
 optimizer: Adam
 beta1: 0.9
 beta2: 0.999
 eps: 1e-08
 amsgrad: True
 milestones: [1000, 1400]
 gamma: 0.1
 save_model: [1000, 1500]
+------------------------------+
Data Get fold 1
Elapsed [1 day, 16:51:16], Epoch [1596/1600], Train_loss = 0.5357, Val_loss = 0.7531, Accuracy = 83.7500
Elapsed [1 day, 16:52:43], Epoch [1597/1600], Train_loss = 0.5312, Val_loss = 0.7618, Accuracy = 83.2500
Elapsed [1 day, 16:53:59], Epoch [1598/1600], Train_loss = 0.5435, Val_loss = 0.7642, Accuracy = 83.0000
Elapsed [1 day, 16:55:16], Epoch [1599/1600], Train_loss = 0.5382, Val_loss = 0.7714, Accuracy = 83.0000
Elapsed [1 day, 16:56:33], Epoch [1600/1600], Train_loss = 0.5351, Val_loss = 0.7648, Accuracy = 83.2500
```

EnvNet4_5

ESC50 数据集 fold 1 结果 **Max Accuracy = 89.0**

Max Accuracy = 89.00 Max Accuracy = 86.75 Max Accuracy = 87.50 Max Accuracy = 90.00 Max Accuracy = 87.00

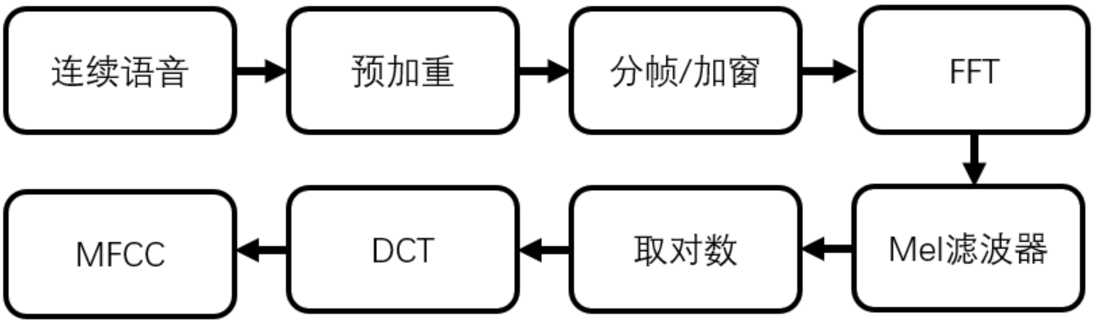
平均准确率：**Mean Accuracy = 88.05**

```
+------------------------------+
| Sound classification
+------------------------------+
 dataset: esc50
 netType: EnvNet4_5
 learning: BC
 augment: True
 nEpochs: 1200
 LRInit: 0.001
 batchSize: 16
 optimizer: Adam
 beta1: 0.9
 beta2: 0.999
 eps: 1e-08
 amsgrad: True
 milestones: [900]
 gamma: 0.1
 save_model: [1000, 1200]
+------------------------------+
Data Get fold 1
Elapsed [4 days, 12:38:57], Epoch [1196/1200], Train_loss = 0.5839, Val_loss = 0.7533, Accuracy = 85.5000
Elapsed [4 days, 12:44:27], Epoch [1197/1200], Train_loss = 0.6015, Val_loss = 0.7392, Accuracy = 85.0000
Elapsed [4 days, 12:49:56], Epoch [1198/1200], Train_loss = 0.5766, Val_loss = 0.7385, Accuracy = 85.2500
Elapsed [4 days, 12:55:25], Epoch [1199/1200], Train_loss = 0.5663, Val_loss = 0.7285, Accuracy = 85.7500
Elapsed [4 days, 13:00:55], Epoch [1200/1200], Train_loss = 0.5758, Val_loss = 0.7280, Accuracy = 85.000
```

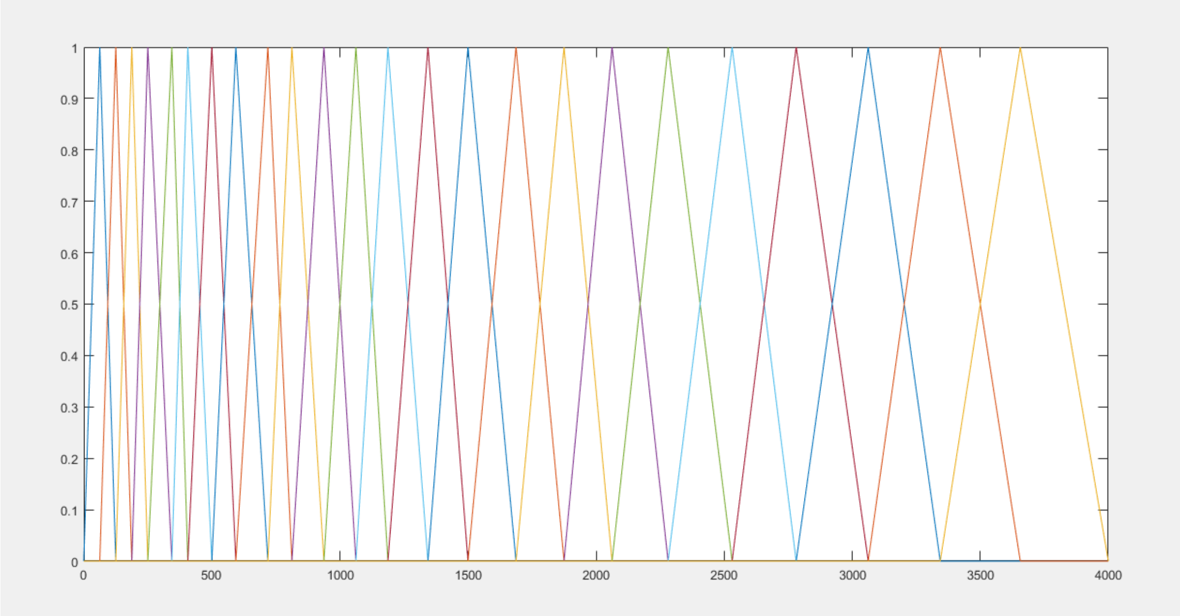
6.Spectrum 调研

logMel调研

Mel特征 提取过程

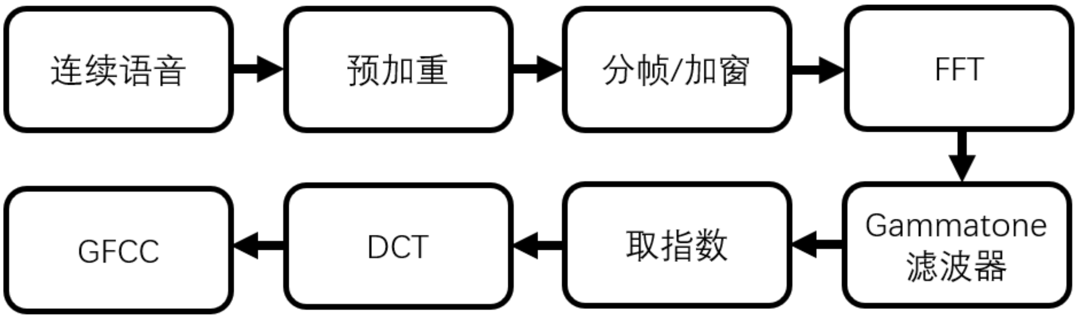


Mel滤波器

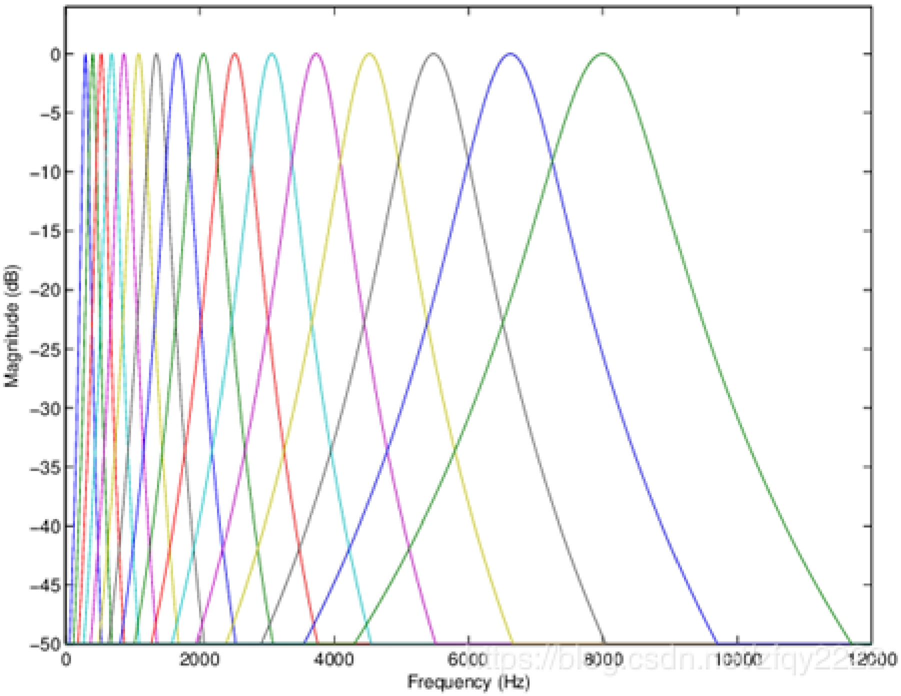


Gammatone 调研

Gammatone 特征提取过程



Gammatone 滤波器



7.EnvNet5

输入特征改为 spectrum

```
Envnet5_1_1      spafe-->logMel

Envnet5_1_2      librosa-->logMel

Envnet5_1_3      librosa-->logMel-->standardization(单通道标准化)

Envnet5_1_4      librosa-->logMel-->delta1-->delta2-->standardization（单通道标准化）

Envnet5_2_1      spafe-->gt

Envnet5_2_2      spafe-->gt-->standardization（单通道标准化）

Envnet5_2_3      spafe-->gfcc

Envnet5_2_4      librosa-->logMel + spafe-->gt -->standardization(单通道标准化)
```

EnvNet5 网络结构

```
class EnvNet5(nn.Module):
    def __init__(self, n_classes):
        super(EnvNet5, self).__init__()
        self.model = nn.Sequential(OrderedDict([

            ('densenet', DenseNet(growth_rate=32, block_config=(6, 12, 24, 16))), #[b, 1024,5,5]

            ('senet1', SEBottleneck(inplanes=256, planes=256//4)),
            ('senet2', SEBottleneck(inplanes=512, planes=512//4)),
            ('global_avg_pool2', nn.AdaptiveAvgPool2d(1)),

            ('flatten', Flatten()),
            ('fc', nn.Linear(in_features=(256+512+1024), out_features=n_classes, bias=True))#[b, n]
            ]))

    def forward(self, x):
        # x [b, 1, 150, 128]
        out1, out2, out3 = self.model[0](x)


        out1 = self.model[3](out1)

        out2 = self.model[1](out2)
        out2 = self.model[3](out2)

        out3 = self.model[2](out3)
        out3 = self.model[3](out3)

        out = torch.cat((out1, out2, out3), dim=1)

        out = self.model[4](out)
        out = self.model[5](out)

        return out
```

特征提取代码

```
from spafe.features import mfcc, gfcc
import librosa

_, log_mel1, _ = mfcc.mfcc(sound, fs=44100, win_len=0.02, win_hop=0.01, nfilts=128)
_, gt, _ = gfcc.gfcc(sound, fs=44100, win_len=0.02, win_hop=0.01, nfilts=128)

mel = librosa.feature.melspectrogram(sound, sr=44100, n_fft=888, hop_length=445, n_mels= 128)
log_mel2 = librosa.power_to_db(mel)
```

EvnNet5_1_1 **spafe --> logMel**

ESC50 数据集 fold 1-2结果 **Max Accuracy = 88.25**   **Max Accuracy = 89.25**

```
+------------------------------+
| Sound classification
+------------------------------+
 dataset: esc50
 netType: EvnNet5_1_1
 learning: BC
 augment: True
 nEpochs: 2000
 LRInit: 0.001
 batchSize: 16
 optimizer: Adam
  beta1: 0.9
 beta2: 0.999
 eps: 1e-08
 amsgrad: True
+------------------------------+
Data Get fold 1
Elapsed [2 days, 9:45:11], Epoch [1996/2000], Train_loss = 0.3221, Val_loss = 0.6944, Accuracy = 84.2500
Elapsed [2 days, 9:46:47], Epoch [1997/2000], Train_loss = 0.3331, Val_loss = 0.6694, Accuracy = 84.5000
Elapsed [2 days, 9:48:22], Epoch [1998/2000], Train_loss = 0.3196, Val_loss = 0.6473, Accuracy = 84.7500
Elapsed [2 days, 9:49:58], Epoch [1999/2000], Train_loss = 0.3191, Val_loss = 0.6599, Accuracy = 85.2500
Elapsed [2 days, 9:51:33], Epoch [2000/2000], Train_loss = 0.3196, Val_loss = 0.6633, Accuracy = 86.2500
+------------------------------+
Data Get fold 2
Elapsed [2 days, 5:44:36], Epoch [1996/2000], Train_loss = 0.3162, Val_loss = 0.6820, Accuracy = 82.0000
Elapsed [2 days, 5:46:12], Epoch [1997/2000], Train_loss = 0.3188, Val_loss = 0.6014, Accuracy = 87.0000
Elapsed [2 days, 5:47:49], Epoch [1998/2000], Train_loss = 0.3064, Val_loss = 0.6272, Accuracy = 84.7500
Elapsed [2 days, 5:49:26], Epoch [1999/2000], Train_loss = 0.3187, Val_loss = 0.6156, Accuracy = 87.7500
Elapsed [2 days, 5:51:02], Epoch [2000/2000], Train_loss = 0.3111, Val_loss = 0.6401, Accuracy = 85.5000
```

EvnNet5_1_2 **librosa --> logMel**

ESC50 数据集 结果

Max Accuracy = 90.50 Max Accuracy = 90.75 Max Accuracy = 91.00 Max Accuracy = 92..50 Max Accuracy = 90.5

平均准确率：**Mean Accuracy = 91.05**

```
+------------------------------+
| Sound classification
+------------------------------+
 dataset: esc50
 netType: EnvNet5_1_2
 learning: BC
 augment: True
 nEpochs: 1000
 LRInit: 0.001
 batchSize: 16
 optimizer: Adam
  beta1: 0.9
 beta2: 0.999
 eps: 1e-08
 amsgrad: True
 milestones: [600, 800]
 gamma: 0.1
+------------------------------+
Data Get fold 1
Elapsed [3 days, 3:42:06], Epoch [996/1000], Train_loss = 0.2901, Val_loss = 0.5794, Accuracy = 88.7500
Elapsed [3 days, 3:46:58], Epoch [997/1000], Train_loss = 0.2879, Val_loss = 0.5863, Accuracy = 88.0000
Elapsed [3 days, 3:51:54], Epoch [998/1000], Train_loss = 0.2986, Val_loss =
0.5773, Accuracy = 89.2500
Elapsed [3 days, 3:56:49], Epoch [999/1000], Train_loss = 0.3031, Val_loss =
0.5713, Accuracy = 89.2500
Elapsed [3 days, 4:01:44], Epoch [1000/1000], Train_loss = 0.3108, Val_loss =
0.5621, Accuracy = 88.0000
+------------------------------+
Data Get fold 2
Elapsed [3 days, 10:36:51], Epoch [996/1000], Train_loss = 0.3239, Val_loss =
0.5445, Accuracy = 87.5000
Elapsed [3 days, 10:41:23], Epoch [997/1000], Train_loss = 0.3118, Val_loss =
0.5631, Accuracy = 87.0000
Elapsed [3 days, 10:45:54], Epoch [998/1000], Train_loss = 0.2962, Val_loss =
0.5537, Accuracy = 86.2500
Elapsed [3 days, 10:50:26], Epoch [999/1000], Train_loss = 0.3055, Val_loss =
0.5592, Accuracy = 86.5000
Elapsed [3 days, 10:54:57], Epoch [1000/1000], Train_loss = 0.3024, Val_loss =
0.5640, Accuracy = 86.0000
+------------------------------+
Data Get fold 3
Elapsed [4 days, 0:15:34], Epoch [996/1000], Train_loss = 0.3376, Val_loss =
0.5066, Accuracy = 90.2500
Elapsed [4 days, 0:21:54], Epoch [997/1000], Train_loss = 0.3498, Val_loss =
0.5068, Accuracy = 90.0000
Elapsed [4 days, 0:28:09], Epoch [998/1000], Train_loss = 0.3574, Val_loss =
0.5040, Accuracy = 89.2500
Elapsed [4 days, 0:34:19], Epoch [999/1000], Train_loss = 0.3253, Val_loss =
0.5130, Accuracy = 89.0000
Elapsed [4 days, 0:40:24], Epoch [1000/1000], Train_loss = 0.3505, Val_loss =
0.4982, Accuracy = 89.5000
+------------------------------+
Data Get fold 4
Elapsed [3 days, 23:51:47], Epoch [996/1000], Train_loss = 0.3467, Val_loss =
0.4307, Accuracy = 91.2500
Elapsed [3 days, 23:57:13], Epoch [997/1000], Train_loss = 0.3509, Val_loss =
0.4296, Accuracy = 91.7500
Elapsed [4 days, 0:02:35], Epoch [998/1000], Train_loss = 0.3463, Val_loss =
0.4294, Accuracy = 91.0000
Elapsed [4 days, 0:08:10], Epoch [999/1000], Train_loss = 0.3655, Val_loss =
0.4205, Accuracy = 91.2500
Elapsed [4 days, 0:13:43], Epoch [1000/1000], Train_loss = 0.3607, Val_loss =
0.4236, Accuracy = 91.2500
+------------------------------+
Data Get fold 5
Elapsed [3 days, 18:38:38], Epoch [996/1000], Train_loss = 0.3387, Val_loss =
0.5119, Accuracy = 89.0000
Elapsed [3 days, 18:43:13], Epoch [997/1000], Train_loss = 0.3437, Val_loss =
0.5083, Accuracy = 88.7500
Elapsed [3 days, 18:47:49], Epoch [998/1000], Train_loss = 0.3669, Val_loss =
0.5017, Accuracy = 90.0000
Elapsed [3 days, 18:52:24], Epoch [999/1000], Train_loss = 0.3402, Val_loss =
0.5112, Accuracy = 89.2500
Elapsed [3 days, 18:56:58], Epoch [1000/1000], Train_loss = 0.3287, Val_loss =
0.5071, Accuracy = 88.7500
```

EvnNet5_1_3 **librosa --> logMel + standardization**

ESC50 数据集 fold 1结果   **Max Accuracy = 89.75**

```
+-------------------------------+
| Sound classification
+-------------------------------+
 dataset: esc50
 netType: EvnNet5_1_3
 learning: BC
 augment: True
 nEpochs: 2000
 LRInit: 0.001
 batchSize: 16
 optimizer: Adam
  beta1: 0.9
 beta2: 0.999
 eps: 1e-08
 amsgrad: True
+-------------------------------+
Data Get fold 1
Elapsed [2 days, 23:07:00], Epoch [1596/1600], Train_loss = 0.2794, Val_loss =
0.6070, Accuracy = 86.7500
Elapsed [2 days, 23:09:35], Epoch [1597/1600], Train_loss = 0.2904, Val_loss =
0.5872, Accuracy = 86.0000
Elapsed [2 days, 23:12:06], Epoch [1598/1600], Train_loss = 0.3084, Val_loss =
0.5862, Accuracy = 86.2500
Elapsed [2 days, 23:14:40], Epoch [1599/1600], Train_loss = 0.2971, Val_loss =
0.5792, Accuracy = 85.5000
Elapsed [2 days, 23:16:49], Epoch [1600/1600], Train_loss = 0.3058, Val_loss =
0.5644, Accuracy = 87.5000
```

EvnNet5_1_4 **librosa --> logMel -->delta1 -->delta2 + standardization**

Max Accuracy = 89.50 Max Accuracy = 87.75 Max Accuracy = 91.50 Max Accuracy = 91.50 Max Accuracy = 88.5

平均准确率：**Mean Accuracy = 89.75**

```
+-------------------------------+
| Sound classification
+-------------------------------+
 dataset: esc50
 netType: EnvNet5_1_4
 learning: BC
 augment: True
 nEpochs: 1000
 LRInit: 0.001
 batchSize: 16
 optimizer: Adam
 beta1: 0.9
 beta2: 0.999
 eps: 1e-08
 amsgrad: True
 milestones: [700, 900]
 gamma: 0.1
 save_model: [500, 800]
+-------------------------------+
Data Get fold 1
Elapsed [1 day, 9:04:19], Epoch [996/1000], Train_loss = 0.7555, Val_loss =
0.6399, Accuracy = 88.5000
```

```
Elapsed [1 day, 9:06:18], Epoch [997/1000], Train_loss = 0.8025, Val_loss =
0.6346, Accuracy = 88.0000
Elapsed [1 day, 9:08:19], Epoch [998/1000], Train_loss = 0.8036, Val_loss =
0.6394, Accuracy = 86.7500
Elapsed [1 day, 9:10:17], Epoch [999/1000], Train_loss = 0.7662, Val_loss =
0.6383, Accuracy = 88.2500
Elapsed [1 day, 9:12:17], Epoch [1000/1000], Train_loss = 0.7507, Val_loss =
0.6450, Accuracy = 88.7500
+------------------------------+
Data Get fold 2
Elapsed [1 day, 5:32:14], Epoch [996/1000], Train_loss = 0.8191, Val_loss =
0.6556, Accuracy = 85.0000
Elapsed [1 day, 5:34:01], Epoch [997/1000], Train_loss = 0.7956, Val_loss =
0.6597, Accuracy = 85.7500
Elapsed [1 day, 5:35:49], Epoch [998/1000], Train_loss = 0.8035, Val_loss =
0.6485, Accuracy = 86.0000
Elapsed [1 day, 5:37:36], Epoch [999/1000], Train_loss = 0.8257, Val_loss =
0.6647, Accuracy = 85.7500
Elapsed [1 day, 5:39:24], Epoch [1000/1000], Train_loss = 0.7951, Val_loss =
0.6447, Accuracy = 86.0000
+------------------------------+
Data Get fold 3
Elapsed [1 day, 6:10:22], Epoch [996/1000], Train_loss = 0.7522, Val_loss =
0.5631, Accuracy = 90.5000
Elapsed [1 day, 6:12:10], Epoch [997/1000], Train_loss = 0.7659, Val_loss =
0.5480, Accuracy = 90.5000
Elapsed [1 day, 6:13:58], Epoch [998/1000], Train_loss = 0.7708, Val_loss =
0.5459, Accuracy = 90.0000
Elapsed [1 day, 6:15:47], Epoch [999/1000], Train_loss = 0.7951, Val_loss =
0.5558, Accuracy = 90.0000
Elapsed [1 day, 6:17:35], Epoch [1000/1000], Train_loss = 0.7452, Val_loss =
0.5479, Accuracy = 89.7500
+------------------------------+
Data Get fold 4
Elapsed [1 day, 5:50:49], Epoch [996/1000], Train_loss = 0.8041, Val_loss =
0.5135, Accuracy = 90.2500
Elapsed [1 day, 5:52:36], Epoch [997/1000], Train_loss = 0.8149, Val_loss =
0.5213, Accuracy = 89.7500
Elapsed [1 day, 5:54:24], Epoch [998/1000], Train_loss = 0.7912, Val_loss =
0.5124, Accuracy = 89.5000
Elapsed [1 day, 5:56:11], Epoch [999/1000], Train_loss = 0.8038, Val_loss =
0.5205, Accuracy = 89.7500
Elapsed [1 day, 5:58:00], Epoch [1000/1000], Train_loss = 0.8145, Val_loss =
0.5181, Accuracy = 90.5000
+------------------------------+
Data Get fold 5
Elapsed [1 day, 6:05:47], Epoch [996/1000], Train_loss = 0.7723, Val_loss =
0.6750, Accuracy = 86.2500
Elapsed [1 day, 6:07:36], Epoch [997/1000], Train_loss = 0.7963, Val_loss =
0.6773, Accuracy = 87.2500
Elapsed [1 day, 6:09:25], Epoch [998/1000], Train_loss = 0.7860, Val_loss =
0.6814, Accuracy = 86.2500
Elapsed [1 day, 6:11:14], Epoch [999/1000], Train_loss = 0.7564, Val_loss =
0.6829, Accuracy = 86.0000
Elapsed [1 day, 6:13:03], Epoch [1000/1000], Train_loss = 0.7804, Val_loss =
0.6745, Accuracy = 86.7500
```

EvnNet5_2_1 **spafe --> gt**

ESC50 数据集 fold 1结果 **Max Accuracy =87.25**

```
+------------------------------+
| Sound classification
+------------------------------+
 dataset: esc50
 netType: EvnNet5_2_1
 learning: BC
 augment: True
 nEpochs: 2000
 LRInit: 0.001
 batchSize: 16
 optimizer: Adam
  beta1: 0.9
 beta2: 0.999
 eps: 1e-08
 amsgrad: True
+------------------------------+
Data Get fold 1
Elapsed [3 days, 3:56:17], Epoch [1596/1600], Train_loss = 0.3704, Val_loss =
0.6882, Accuracy = 85.7500
Elapsed [3 days, 3:58:48], Epoch [1597/1600], Train_loss = 0.3528, Val_loss =
0.6709, Accuracy = 85.7500
Elapsed [3 days, 4:01:29], Epoch [1598/1600], Train_loss = 0.3604, Val_loss =
0.7113, Accuracy = 83.7500
Elapsed [3 days, 4:03:55], Epoch [1599/1600], Train_loss = 0.3686, Val_loss =
0.6535, Accuracy = 86.5000
Elapsed [3 days, 4:06:25], Epoch [1600/1600], Train_loss = 0.3397, Val_loss =
0.6956, Accuracy = 83.5000
```

EnvNet5_2_2 **spafe --> gt + standardization**

ESC50 数据集 fold 1结果 **Max Accuracy = 86.75**

```
+------------------------------+
| Sound classification
+------------------------------+
 dataset: esc50
 netType: EnvNet5_2_2
 learning: BC
 augment: True
 nEpochs: 2000
 LRInit: 0.001
 batchSize: 16
 optimizer: Adam
  beta1: 0.9
 beta2: 0.999
 eps: 1e-08
 amsgrad: True
+------------------------------+
Data Get fold 1
Elapsed [2 days, 18:53:34], Epoch [1496/2000], Train_loss = 0.3746, Val_loss =
0.7132, Accuracy = 82.0000
Elapsed [2 days, 18:56:50], Epoch [1497/2000], Train_loss = 0.3613, Val_loss =
0.7291, Accuracy = 82.7500
```

```
Elapsed [2 days, 19:00:00], Epoch [1498/2000], Train_loss = 0.3728, Val_loss = 0.7633, Accuracy = 81.2500
Elapsed [2 days, 19:03:08], Epoch [1499/2000], Train_loss = 0.3764, Val_loss = 0.8036, Accuracy = 81.7500
Elapsed [2 days, 19:06:21], Epoch [1500/2000], Train_loss = 0.3605, Val_loss = 0.7488, Accuracy = 83.2500
```

EnvNet5_2_3 **spafe --> gfcc**

ESC50 数据集 fold 1结果 **Max Accuracy = 85.00**

```
+------------------------------+
| Sound classification
+------------------------------+
 dataset: esc50
 netType: EnvNet5_2_3
 learning: BC
 augment: True
 nEpochs: 1000
 LRInit: 0.001
 batchSize: 16
 optimizer: Adam
  beta1: 0.9
 beta2: 0.999
 eps: 1e-08
 amsgrad: True
+------------------------------+
Data Get fold 1
Elapsed [1 day, 6:19:31], Epoch [996/1000], Train_loss = 0.6733, Val_loss = 0.8389, Accuracy = 81.7500
Elapsed [1 day, 6:21:21], Epoch [997/1000], Train_loss = 0.6893, Val_loss = 0.7944, Accuracy = 84.2500
Elapsed [1 day, 6:23:11], Epoch [998/1000], Train_loss = 0.6854, Val_loss = 0.8007, Accuracy = 81.7500
Elapsed [1 day, 6:25:01], Epoch [999/1000], Train_loss = 0.7027, Val_loss = 0.8362, Accuracy = 83.0000
Elapsed [1 day, 6:26:51], Epoch [1000/1000], Train_loss = 0.6888, Val_loss = 0.8938, Accuracy = 81.2500
```

EnvNet5_2_4 librosa-->logMel + spafe-->gt -->standardization(单通道标准化)

Max Accuracy = 89.50 **Max Accuracy = 89.25** **Max Accuracy = 90.50** **Max Accuracy = 92.00** **Max Accuracy = 87.5**

平均准确率：**Mean Accuracy = 89.75**

```
+------------------------------+
| Sound classification
+------------------------------+
 dataset: esc50
 netType: EnvNet5_2_4
 learning: BC
 augment: True
 nEpochs: 1000
 LRInit: 0.001
 batchSize: 16
 optimizer: Adam
 beta1: 0.9
 beta2: 0.999
 eps: 1e-08
 amsgrad: True
 milestones: [700, 900]
 gamma: 0.1
 save_model: [500, 800]
+------------------------------+
Data Get fold 1
Elapsed [1 day, 11:50:19], Epoch [996/1000], Train_loss = 0.7867, Val_loss =
0.6707, Accuracy = 87.2500
Elapsed [1 day, 11:52:28], Epoch [997/1000], Train_loss = 0.7971, Val_loss =
0.6428, Accuracy = 88.5000
Elapsed [1 day, 11:54:37], Epoch [998/1000], Train_loss = 0.8031, Val_loss =
0.6498, Accuracy = 87.5000
Elapsed [1 day, 11:56:47], Epoch [999/1000], Train_loss = 0.8046, Val_loss =
0.6572, Accuracy = 87.7500
Elapsed [1 day, 11:58:56], Epoch [1000/1000], Train_loss = 0.7831, Val_loss =
0.6542, Accuracy = 87.2500
+------------------------------+
Data Get fold 2
Elapsed [1 day, 12:08:50], Epoch [996/1000], Train_loss = 0.7797, Val_loss =
0.6489, Accuracy = 85.2500
Elapsed [1 day, 12:11:00], Epoch [997/1000], Train_loss = 0.7895, Val_loss =
0.6468, Accuracy = 85.7500
Elapsed [1 day, 12:13:10], Epoch [998/1000], Train_loss = 0.7722, Val_loss =
0.6405, Accuracy = 87.0000
Elapsed [1 day, 12:15:20], Epoch [999/1000], Train_loss = 0.8234, Val_loss =
0.6493, Accuracy = 86.2500
Elapsed [1 day, 12:17:30], Epoch [1000/1000], Train_loss = 0.8023, Val_loss =
0.6439, Accuracy = 86.5000
+------------------------------+
Data Get fold 3
Elapsed [1 day, 12:09:27], Epoch [996/1000], Train_loss = 0.8177, Val_loss =
0.5987, Accuracy = 88.7500
Elapsed [1 day, 12:11:38], Epoch [997/1000], Train_loss = 0.7801, Val_loss =
0.6050, Accuracy = 89.2500
Elapsed [1 day, 12:13:48], Epoch [998/1000], Train_loss = 0.8110, Val_loss =
0.6109, Accuracy = 88.7500
Elapsed [1 day, 12:15:59], Epoch [999/1000], Train_loss = 0.8067, Val_loss =
0.6058, Accuracy = 88.7500
Elapsed [1 day, 12:18:09], Epoch [1000/1000], Train_loss = 0.8023, Val_loss =
0.6100, Accuracy = 88.7500
+------------------------------+
Data Get fold 4
Elapsed [1 day, 12:29:02], Epoch [996/1000], Train_loss = 0.8149, Val_loss =
0.5510, Accuracy = 90.0000
Elapsed [1 day, 12:31:15], Epoch [997/1000], Train_loss = 0.8082, Val_loss =
0.5573, Accuracy = 89.7500
Elapsed [1 day, 12:33:28], Epoch [998/1000], Train_loss = 0.8154, Val_loss =
0.5515, Accuracy = 89.5000
Elapsed [1 day, 12:35:40], Epoch [999/1000], Train_loss = 0.8199, Val_loss =
0.5496, Accuracy = 89.7500
Elapsed [1 day, 12:37:52], Epoch [1000/1000], Train_loss = 0.8007, Val_loss =
0.5553, Accuracy = 89.2500
+------------------------------+
Data Get fold 5
```

```
Elapsed [1 day, 12:23:19], Epoch [996/1000], Train_loss = 0.7797, Val_loss =
0.6883, Accuracy = 85.0000
Elapsed [1 day, 12:25:31], Epoch [997/1000], Train_loss = 0.7970, Val_loss =
0.6824, Accuracy = 85.2500
Elapsed [1 day, 12:27:43], Epoch [998/1000], Train_loss = 0.7880, Val_loss =
0.6851, Accuracy = 85.7500
Elapsed [1 day, 12:29:55], Epoch [999/1000], Train_loss = 0.7702, Val_loss =
0.6802, Accuracy = 85.5000
Elapsed [1 day, 12:32:07], Epoch [1000/1000], Train_loss = 0.7857, Val_loss =
0.7017, Accuracy = 86.2500
```

8.SpecNet2

Environment Sound Classification using Multiple Feature Channels and Deep Convolutional Neural Networks

```
https://arxiv.org/abs/1908.11219
```

双向卷积 加入 Multi-Scale + Mish + Global Avg Pooling

model

```
class SpecNet2(nn.Module):
    def __init__(self, n_classes=50, out_ch=32):
        super(SpecNet2, self).__init__()

        self.first_conv = nn.Sequential(
            nn.Conv2d(in_channels=1, out_channels=out_ch, kernel_size=3,
stride=1, padding=1, bias=False),
            nn.BatchNorm2d(out_ch),
            Mish(),
        )
        self.conv2 = nn.Sequential(
            Conv_Block(out_ch, out_ch, (1,3), (1,1), (0,1)),
            MaxPool((1,3), (1,2),(0,1))
        )
        self.conv3 = nn.Sequential(
            Conv_Block(out_ch, out_ch*2, (5,1), (1,1), (2,0)),
            MaxPool((3,1), (2,1), (1,0)),
        )
        self.conv4 = nn.Sequential(
            Conv_Block(out_ch*2, out_ch*2, (1, 3), (1, 1), (0, 1)),
            MaxPool((1, 3), (1, 2), (0, 1))
        )
        self.conv5 = nn.Sequential(
            Conv_Block(out_ch*2, out_ch*4, (5, 1), (1, 1), (2, 0)),
            MaxPool((3, 1), (2, 1), (1, 0)),
        )
        self.conv6 = nn.Sequential(
            Conv_Block(out_ch*4, out_ch*4, (1, 3), (1, 1), (0, 1)),
            MaxPool((1, 3), (1, 2), (0, 1))
        )
        self.conv7 = nn.Sequential(
            Conv_Block(out_ch*4, out_ch*8, (5, 1), (1, 1), (2, 0)),
            MaxPool((3, 1), (2, 1), (1, 0)),
        )
        self.conv8 = nn.Sequential(
            Conv_Block(out_ch*8, out_ch*16, (5, 3), (1, 1), (2, 1)),
            MaxPool((3, 3), (2, 2), (1, 1))
        )
        self.conv9 = nn.Sequential(
            Conv_Block(out_ch*16, out_ch*32, (5, 3), (1, 1), (2, 1)),
            MaxPool((3, 3), (2, 2), (1, 1)),
        )
        self.linear = nn.Sequential(
            nn.Linear(256 + 512 + 1024, 512),
            Mish(),
            nn.Dropout(p=0.5),
            nn.Linear(512, n_classes),
        )
        self.global_avg_pool = nn.AdaptiveAvgPool2d(1)


    def forward(self, x):
        # x [b, 1, 150, 128]
        x1 = self.first_conv(x)
        x2 = self.conv2(x1)
        x3 = self.conv3(x2)
        x4 = self.conv4(x3)
        x5 = self.conv5(x4)
        x6 = self.conv6(x5)
        x7 = self.conv7(x6)
        x8 = self.conv8(x7)
        x9 = self.conv9(x8)

        out1 = self.global_avg_pool(x7)
        out2 = self.global_avg_pool(x8)
        out3 = self.global_avg_pool(x9)

        out = torch.cat((out1, out2, out3), dim=1)
        out = out.view(out.size(0), -1)
        out = self.linear(out)

        return out
```

SpecNet2 **librosa --> logMel**

ESC50 数据集 fold 1结果 **Max Accuracy = 89.75**

```
+------------------------------+
| Sound classification
+------------------------------+
 dataset: esc50
 netType: SpecNet2
 learning: BC
 augment: True
 nEpochs: 2000
 LRInit: 0.001
 batchSize: 16
 optimizer: Adam
  beta1: 0.9
 beta2: 0.999
 eps: 1e-08
 amsgrad: True
+-----------------------------+
Data Get fold 1
Elapsed [2 days, 11:17:25], Epoch [1496/2000], Train_loss = 0.2964, Val_loss =
0.6351, Accuracy = 87.2500
Elapsed [2 days, 11:20:15], Epoch [1497/2000], Train_loss = 0.2946, Val_loss =
0.6279, Accuracy = 85.7500
Elapsed [2 days, 11:23:05], Epoch [1498/2000], Train_loss = 0.2866, Val_loss =
0.6088, Accuracy = 86.7500
Elapsed [2 days, 11:25:57], Epoch [1499/2000], Train_loss = 0.2889, Val_loss =
0.6156, Accuracy = 85.0000
Elapsed [2 days, 11:28:50], Epoch [1500/2000], Train_loss = 0.2907, Val_loss =
0.6091, Accuracy = 87.5000
```

9.EnvNet6

Envnet5模型中的 Densenet 模块 卷积 变为 双向卷积

```
1:在denseblock内部convolution方式

    1a: 1*3 → 3*1 → 1*3 → 3*1 → …

    1b  1*3 → 1*3 → 1*3 → … → 3*1 → 3*1 → 3*1 → …

    1c: 1*3 → 1*3 → 1*3… (3*1 → 3*1 → 3*1…)

2:在denseblock之间

    2a: denseblock 只在一个方向做 （transition block也需要做相应调整）

    2b: denseblock 在两个方向上做

3:4个denseblock

    3a: 全部双向conv

    3b: 前面两个denseblock双向conv，后面两个做标准3*3

    3c: 前面三个denseblock双向conv，后面一个做标准3*3

    3d: 只有第一个denseblock双向conv，后面三个都做标准的3*3

Envnet 6_1_1：1a+2b+3c

Envnet 6_1_2：1b+2b+3c

Envnet 6_1_3：1b+2b+3d

Envnet 6_2_1：1c+2a+3c

Envnet 6_3_1  Envnet 6.1.1 + logMel + gt +  数据集 standardization

Envnet 6_3_2  Envnet 6.1.1 + logMel + gt +  normalization(单通道)

Envnet 6_4_1  Envnet 6.1.1 + logMel + Urbansound8k

Envnet 6_4_2 Envnet 6.1.1 + gt + Urbansound8k + normalization(单通道)
Envnet 6_4_3 Envnet 6.1.1 + gt + Urbansound8k

Envnet 6_5_1 Envnet 6.1.1 + DenseNet(growth_rate=16, block_config=(6, 12, 24,
16)))
```

Envnet 6_1_1

```
DenseNet2(growth_rate=32, block_config=(6, 12, 24, 16))
#6, 12 --> 前两个block双向1*3+3*1{6}+1*3+3*1{12}
```

ESC50 数据集 结果

Max Accuracy = 90.50 Max Accuracy = 91.75 Max Accuracy = 90.75 Max Accuracy = 93.50 Max Accuracy = 90.25

平均准确率：**Mean Accuracy = 91.35**

```
+------------------------------+
| Sound classification
+------------------------------+
 dataset: esc50
 netType: EnvNet6_1_1
 learning: BC
 augment: True
 nEpochs: 1000
 LRInit: 0.001
 batchSize: 16
 optimizer: Adam
  beta1: 0.9
 beta2: 0.999
 eps: 1e-08
 amsgrad: True
 milestones: [800]
 gamma: 0.1
+------------------------------+
Data Get fold 1
Elapsed [1 day, 2:04:05], Epoch [996/1000], Train_loss = 0.3106, Val_loss =
0.5237, Accuracy = 88.7500
Elapsed [1 day, 2:05:39], Epoch [997/1000], Train_loss = 0.3150, Val_loss =
0.5333, Accuracy = 89.0000
Elapsed [1 day, 2:07:13], Epoch [998/1000], Train_loss = 0.3171, Val_loss =
0.5434, Accuracy = 89.0000
Elapsed [1 day, 2:08:47], Epoch [999/1000], Train_loss = 0.3086, Val_loss =
0.5345, Accuracy = 89.5000
Elapsed [1 day, 2:10:22], Epoch [1000/1000], Train_loss = 0.3138, Val_loss =
0.5431, Accuracy = 88.0000
+------------------------------+
Data Get fold 2
Elapsed [3 days, 23:41:17], Epoch [996/1000], Train_loss = 0.2975, Val_loss =
0.5372, Accuracy = 88.7500
Elapsed [3 days, 23:48:57], Epoch [997/1000], Train_loss = 0.2988, Val_loss =
0.5448, Accuracy = 88.0000
Elapsed [3 days, 23:56:32], Epoch [998/1000], Train_loss = 0.2980, Val_loss =
0.5329, Accuracy = 88.7500
Elapsed [4 days, 0:04:08], Epoch [999/1000], Train_loss = 0.3151, Val_loss =
0.5225, Accuracy = 88.7500
```

```
Elapsed [4 days, 0:11:44], Epoch [1000/1000], Train_loss = 0.3029, Val_loss =
0.5328, Accuracy = 88.7500
+------------------------------+
Data Get fold 3
Elapsed [3 days, 19:35:25], Epoch [996/1000], Train_loss = 0.3592, Val_loss =
0.5088, Accuracy = 88.5000
Elapsed [3 days, 19:41:05], Epoch [997/1000], Train_loss = 0.3410, Val_loss =
0.4991, Accuracy = 90.0000
Elapsed [3 days, 19:46:43], Epoch [998/1000], Train_loss = 0.3516, Val_loss =
0.5060, Accuracy = 89.2500
Elapsed [3 days, 19:52:14], Epoch [999/1000], Train_loss = 0.3509, Val_loss =
0.5123, Accuracy = 89.7500
Elapsed [3 days, 19:57:52], Epoch [1000/1000], Train_loss = 0.3486, Val_loss =
0.5080, Accuracy = 89.2500
+------------------------------+
Data Get fold 4
Elapsed [3 days, 20:28:27], Epoch [996/1000], Train_loss = 0.3587, Val_loss =
0.4394, Accuracy = 92.5000
Elapsed [3 days, 20:33:23], Epoch [997/1000], Train_loss = 0.3459, Val_loss =
0.5058, Accuracy = 92.0000
Elapsed [3 days, 20:38:16], Epoch [998/1000], Train_loss = 0.3539, Val_loss =
0.4450, Accuracy = 92.7500
Elapsed [3 days, 20:43:10], Epoch [999/1000], Train_loss = 0.3517, Val_loss =
0.4395, Accuracy = 92.7500
Elapsed [3 days, 20:48:06], Epoch [1000/1000], Train_loss = 0.3551, Val_loss =
0.4370, Accuracy = 92.2500
+------------------------------+
Data Get fold 5
Elapsed [3 days, 16:20:00], Epoch [996/1000], Train_loss = 0.3426, Val_loss =
0.5291, Accuracy = 88.2500
Elapsed [3 days, 16:25:04], Epoch [997/1000], Train_loss = 0.3356, Val_loss =
0.5251, Accuracy = 88.0000
Elapsed [3 days, 16:30:14], Epoch [998/1000], Train_loss = 0.3652, Val_loss =
0.5210, Accuracy = 88.5000
Elapsed [3 days, 16:35:25], Epoch [999/1000], Train_loss = 0.3425, Val_loss =
0.5348, Accuracy = 89.0000
Elapsed [3 days, 16:40:46], Epoch [1000/1000], Train_loss = 0.3435, Val_loss =
0.5325, Accuracy = 88.5000
```

Envnet 6_1_2

```
DenseNet2(growth_rate=32, block_config=(6, 12, 24, 16))
#6, 12 --> 前两个block双向1*3{3}+3*1{3}, 1*3{6}+3*1{6}
```

ESC50 数据集 fold 1结果 **Max Accuracy =90.00**

```
+------------------------------+
| Sound classification
+------------------------------+
 dataset: esc50
 netType: EnvNet6_1_2
 learning: BC
 augment: True
 nEpochs: 1600
 LRInit: 0.001
 batchSize: 16
 optimizer: Adam
  beta1: 0.9
 beta2: 0.999
 eps: 1e-08
 amsgrad: True
 milestones: [800]
 gamma: 0.1
+------------------------------+
Data Get fold 1
Elapsed [1 day, 2:11:39], Epoch [996/1600], Train_loss = 0.3195, Val_loss = 
0.5594, Accuracy = 87.2500
Elapsed [1 day, 2:13:14], Epoch [997/1600], Train_loss = 0.3109, Val_loss = 
0.5617, Accuracy = 87.2500
Elapsed [1 day, 2:14:49], Epoch [998/1600], Train_loss = 0.3021, Val_loss = 
0.5623, Accuracy = 88.2500
Elapsed [1 day, 2:16:24], Epoch [999/1600], Train_loss = 0.3058, Val_loss = 
0.5680, Accuracy = 88.5000
Elapsed [1 day, 2:17:59], Epoch [1000/1600], Train_loss = 0.3021, Val_loss = 
0.507, Accuracy = 88.0000
```

Envnet 6_2_1

```
DenseNet2(growth_rate=32, block_config=(6, 6, 12, 12, 24, 16))
#6, 6, 12, 12 -->前四个block双向 1*3{6}, 3*1{6}, 1*3{12}, 3*1{12}
```

ESC50 数据集 fold 1结果 **Max Accuracy =89.00**

```
+------------------------------+
| Sound classification
+------------------------------+
 dataset: esc50
 netType: EnvNet6_2_1
 learning: BC
 augment: True
 nEpochs: 1600
 LRInit: 0.001
 batchSize: 16
 optimizer: Adam
  beta1: 0.9
 beta2: 0.999
 eps: 1e-08
 amsgrad: True
 milestones: [800]
 gamma: 0.1
+------------------------------+
Data Get fold 1
Elapsed [20:38:37], Epoch [780/1600], Train_loss = 0.7221, Val_loss = 0.6992, 
Accuracy = 87.7500
Elapsed [20:40:13], Epoch [781/1600], Train_loss = 0.7531, Val_loss = 0.7059, 
Accuracy = 88.5000
Elapsed [20:41:48], Epoch [782/1600], Train_loss = 0.7421, Val_loss = 0.6914, 
Accuracy = 87.7500
Elapsed [20:43:24], Epoch [783/1600], Train_loss = 0.7410, Val_loss = 0.6951, 
Accuracy = 88.5000
```

```
Elapsed [20:44:59], Epoch [784/1600], Train_loss = 0.7104, Val_loss = 0.7061,
Accuracy = 87.5000
```

Envnet 6_3_1   logMel+gt + 数据集 standardization

ESC50 数据集 fold 1结果   **Max Accuracy =90.00**

```
+------------------------------+
| Sound classification
+------------------------------+
 dataset: esc50
 netType: EnvNet6_3_1
 learning: BC
 augment: True
 nEpochs: 1600
 LRInit: 0.001
 batchSize: 16
 optimizer: Adam
  beta1: 0.9
 beta2: 0.999
 eps: 1e-08
 amsgrad: True
 milestones: [800]
 gamma: 0.1
+------------------------------+
Data Get fold 1
Elapsed [1 day, 11:47:15], Epoch [996/1000], Train_loss = 0.7594, Val_loss =
0.6501, Accuracy = 88.2500
Elapsed [1 day, 11:49:25], Epoch [997/1000], Train_loss = 0.7787, Val_loss =
0.6780, Accuracy = 88.5000
Elapsed [1 day, 11:51:35], Epoch [998/1000], Train_loss = 0.7831, Val_loss =
0.6810, Accuracy = 88.2500
Elapsed [1 day, 11:53:44], Epoch [999/1000], Train_loss = 0.7878, Val_loss =
0.6445, Accuracy = 88.7500
Elapsed [1 day, 11:55:53], Epoch [1000/1000], Train_loss = 0.7673, Val_loss =
0.6479, Accuracy = 88.2500
```

Envnet 6_3_2   logMel+gt + 单通道 归一化

ESC50 数据集 fold 1结果   **Max Accuracy =89.50**

```
+------------------------------+
| Sound classification
+------------------------------+
 dataset: esc50
 netType: EnvNet6_3_2
 learning: BC
 augment: True
 nEpochs: 1600
 LRInit: 0.001
 batchSize: 16
 optimizer: Adam
  beta1: 0.9
 beta2: 0.999
 eps: 1e-08
 amsgrad: True
 milestones: [900]
 gamma: 0.1
+------------------------------+
Data Get fold 1
Elapsed [3:28:34], Epoch [96/1000], Train_loss = 0.3387, Val_loss = 0.5664,
Accuracy = 88.0000
Elapsed [3:30:44], Epoch [97/1000], Train_loss = 0.3542, Val_loss = 0.5826,
Accuracy = 87.2500
Elapsed [3:32:55], Epoch [98/1000], Train_loss = 0.3445, Val_loss = 0.5764,
Accuracy = 87.0000
Elapsed [3:35:06], Epoch [99/1000], Train_loss = 0.3191, Val_loss = 0.5610,
Accuracy = 86.7500
Elapsed [3:37:17], Epoch [100/1000], Train_loss = 0.3270, Val_loss = 0.5660,
Accuracy = 88.0000
```

Envnet 6_4_1

urbansound8k 数据集 结果

Urbansound8k	Envnet 6.4.1(logMel)
fold1	83.9633
fold2	85.0225
fold3	74.9189
fold4	85.2525
fold5	88.9957
fold6	80.4374
fold7	88.5442
fold8	77.1712
fold9	84.6814
fold10	88.0526
mean	**83.70397**

Envnet 6_5_1

Max Accuracy = 89.75   Max Accuracy = 87.5 Max Accuracy = 90.25   Max Accuracy = 92.00 Max Accuracy = 88.25

平均准确率：**Mean Accuracy = 89.55**

```
+------------------------------+
| Sound classification
+------------------------------+
 dataset: esc50
 netType: EnvNet6_5_1
 learning: BC
 augment: True
 nEpochs: 1000
 LRInit: 0.001
 batchSize: 16
 optimizer: Adam
 beta1: 0.9
 beta2: 0.999
 eps: 1e-08
 amsgrad: True
 milestones: [700, 900]
 gamma: 0.1
 save_model: [500, 800]
+------------------------------+
Data Get fold 1
Elapsed [1 day, 4:50:39], Epoch [996/1000], Train_loss = 0.7724, Val_loss =
0.6593, Accuracy = 87.7500
Elapsed [1 day, 4:52:22], Epoch [997/1000], Train_loss = 0.7710, Val_loss =
0.6608, Accuracy = 86.7500
Elapsed [1 day, 4:54:04], Epoch [998/1000], Train_loss = 0.7616, Val_loss =
0.6630, Accuracy = 87.2500
Elapsed [1 day, 4:55:46], Epoch [999/1000], Train_loss = 0.7840, Val_loss =
0.6631, Accuracy = 87.7500
Elapsed [1 day, 4:57:29], Epoch [1000/1000], Train_loss = 0.7837, Val_loss =
0.6525, Accuracy = 87.7500
+------------------------------+
Data Get fold 2
Elapsed [1 day, 5:20:34], Epoch [996/1000], Train_loss = 0.7747, Val_loss =
0.6673, Accuracy = 85.0000
Elapsed [1 day, 5:22:20], Epoch [997/1000], Train_loss = 0.7476, Val_loss =
0.6699, Accuracy = 84.5000
Elapsed [1 day, 5:24:05], Epoch [998/1000], Train_loss = 0.7490, Val_loss =
0.6681, Accuracy = 85.0000
Elapsed [1 day, 5:25:48], Epoch [999/1000], Train_loss = 0.7578, Val_loss =
0.6731, Accuracy = 85.0000
Elapsed [1 day, 5:27:32], Epoch [1000/1000], Train_loss = 0.7753, Val_loss =
0.6620, Accuracy = 85.5000
+------------------------------+
Data Get fold 3
Elapsed [1 day, 1:37:38], Epoch [996/1000], Train_loss = 0.7857, Val_loss =
0.5808, Accuracy = 88.2500
Elapsed [1 day, 1:39:04], Epoch [997/1000], Train_loss = 0.7729, Val_loss =
0.5825, Accuracy = 89.0000
Elapsed [1 day, 1:40:30], Epoch [998/1000], Train_loss = 0.7500, Val_loss =
0.5873, Accuracy = 89.2500
Elapsed [1 day, 1:41:56], Epoch [999/1000], Train_loss = 0.7736, Val_loss =
0.5917, Accuracy = 88.5000
Elapsed [1 day, 1:43:23], Epoch [1000/1000], Train_loss = 0.7819, Val_loss =
0.5960, Accuracy = 89.5000
+------------------------------+
Data Get fold 4
Elapsed [1 day, 0:07:23], Epoch [996/1000], Train_loss = 0.7921, Val_loss =
0.5344, Accuracy = 89.5000
Elapsed [1 day, 0:08:50], Epoch [997/1000], Train_loss = 0.7750, Val_loss =
0.5417, Accuracy = 89.5000
Elapsed [1 day, 0:10:17], Epoch [998/1000], Train_loss = 0.7609, Val_loss =
0.5156, Accuracy = 91.0000
Elapsed [1 day, 0:11:44], Epoch [999/1000], Train_loss = 0.7793, Val_loss =
0.5291, Accuracy = 89.5000
Elapsed [1 day, 0:13:11], Epoch [1000/1000], Train_loss = 0.7718, Val_loss =
0.5357, Accuracy = 90.0000
+------------------------------+
Data Get fold 5
Elapsed [1 day, 0:04:58], Epoch [996/1000], Train_loss = 0.7576, Val_loss =
0.6548, Accuracy = 87.2500
Elapsed [1 day, 0:06:25], Epoch [997/1000], Train_loss = 0.7690, Val_loss =
0.6522, Accuracy = 87.2500
Elapsed [1 day, 0:07:53], Epoch [998/1000], Train_loss = 0.7512, Val_loss =
0.6400, Accuracy = 86.5000
Elapsed [1 day, 0:09:20], Epoch [999/1000], Train_loss = 0.7670, Val_loss =
0.6315, Accuracy = 86.7500
Elapsed [1 day, 0:10:48], Epoch [1000/1000], Train_loss = 0.7504, Val_loss =
0.6391, Accuracy = 86.5000
```

10.Urbansound8k unofficial

all smaples 划分为 20%的验证集 80%的训练集

Envnet6_6_1 **Max Accuracy = 97.9452**

```
+------------------------------+
| Sound classification
+------------------------------+
 dataset: UrbanSound8K
 netType: EnvNet6_6_1
 learning: BC
 augment: True
 nEpochs: 500
 LRInit: 0.001
 batchSize: 16
 optimizer: Adam
 beta1: 0.9
 beta2: 0.999
 eps: 1e-08
 amsgrad: True
 milestones: [200, 400]
 gamma: 0.1
 save_model: [200, 300, 400, 500]
+------------------------------+
Elapsed [2 days, 7:32:04], Epoch [496/500], Train_loss = 0.4543, Val_loss =
0.2351, Accuracy = 97.4886
Elapsed [2 days, 7:38:48], Epoch [497/500], Train_loss = 0.4523, Val_loss =
0.2306, Accuracy = 97.3174
Elapsed [2 days, 7:45:29], Epoch [498/500], Train_loss = 0.4664, Val_loss =
0.2366, Accuracy = 97.4315
Elapsed [2 days, 7:52:11], Epoch [499/500], Train_loss = 0.4533, Val_loss =
0.2253, Accuracy = 97.5457
Elapsed [2 days, 7:58:55], Epoch [500/500], Train_loss = 0.4570, Val_loss =
0.2189, Accuracy = 97.4315
```

Envnet5_3_1 **Max Accuracy = 98.1735**

```
+------------------------------+
| Sound classification
+------------------------------+
 dataset: UrbanSound8K
 netType: EnvNet5_3_1
 learning: BC
 augment: True
 nEpochs: 500
 LRInit: 0.001
 batchSize: 16
 optimizer: Adam
 beta1: 0.9
 beta2: 0.999
 eps: 1e-08
 amsgrad: True
 milestones: [200, 400]
 gamma: 0.1
 save_model: [200, 300, 400, 500]
+------------------------------+
Elapsed [2 days, 7:11:08], Epoch [496/500], Train_loss = 0.4533, Val_loss =
0.2466, Accuracy = 97.8311
Elapsed [2 days, 7:17:48], Epoch [497/500], Train_loss = 0.4641, Val_loss =
0.2229, Accuracy = 97.9452
Elapsed [2 days, 7:24:27], Epoch [498/500], Train_loss = 0.4539, Val_loss =
0.2382, Accuracy = 97.8881
Elapsed [2 days, 7:31:07], Epoch [499/500], Train_loss = 0.4495, Val_loss =
0.2226, Accuracy = 97.7740
Elapsed [2 days, 7:37:47], Epoch [500/500], Train_loss = 0.4608, Val_loss =
0.2222, Accuracy = 98.1735
```

Envnet4_6_1 **Max Accuracy = 92.6941**

```
+------------------------------+
| Sound classification
+------------------------------+
 dataset: UrbanSound8K
 netType: EnvNet4_6_1
 learning: BC
 augment: True
 nEpochs: 500
 LRInit: 0.001
 batchSize: 16
 optimizer: Adam
 beta1: 0.9
 beta2: 0.999
 eps: 1e-08
 amsgrad: True
 milestones: [200, 400]
 gamma: 0.1
 save_model: [200, 300, 400, 500]
+------------------------------+
Elapsed [1 day, 21:05:25], Epoch [496/500], Train_loss = 0.7164, Val_loss =
0.4101, Accuracy = 91.9521
Elapsed [1 day, 21:10:51], Epoch [497/500], Train_loss = 0.7165, Val_loss =
0.4118, Accuracy = 92.2374
Elapsed [1 day, 21:16:18], Epoch [498/500], Train_loss = 0.7079, Val_loss =
0.4255, Accuracy = 91.0388
Elapsed [1 day, 21:21:44], Epoch [499/500], Train_loss = 0.7103, Val_loss =
0.4014, Accuracy = 91.4954
Elapsed [1 day, 21:27:10], Epoch [500/500], Train_loss = 0.7102, Val_loss =
0.3970, Accuracy = 92.4658
```

Envnet3_6_1 **Max Accuracy = 93.0365**

```
+------------------------------+
| Sound classification
+------------------------------+
 dataset: UrbanSound8K
 netType: EnvNet3_6_1
 learning: BC
 augment: True
 nEpochs: 500
 LRInit: 0.001
 batchSize: 16
 optimizer: Adam
 beta1: 0.9
 beta2: 0.999
 eps: 1e-08
 amsgrad: True
 milestones: [200, 400]
 gamma: 0.1
 save_model: [200, 300, 400, 500]
+------------------------------+
Elapsed [1 day, 20:10:57], Epoch [496/500], Train_loss = 0.7014, Val_loss = 0.3943, Accuracy = 92.6941
Elapsed [1 day, 20:16:17], Epoch [497/500], Train_loss = 0.7072, Val_loss = 0.3919, Accuracy = 92.5799
Elapsed [1 day, 20:21:37], Epoch [498/500], Train_loss = 0.7061, Val_loss = 0.3883, Accuracy = 92.8653
Elapsed [1 day, 20:26:57], Epoch [499/500], Train_loss = 0.7091, Val_loss = 0.3965, Accuracy = 93.0365
Elapsed [1 day, 20:32:20], Epoch [500/500], Train_loss = 0.7061, Val_loss = 0.3944, Accuracy = 92.2945
```