Sound Event Detection of Weakly Labelled Data

zhitiankai

2020.11.30

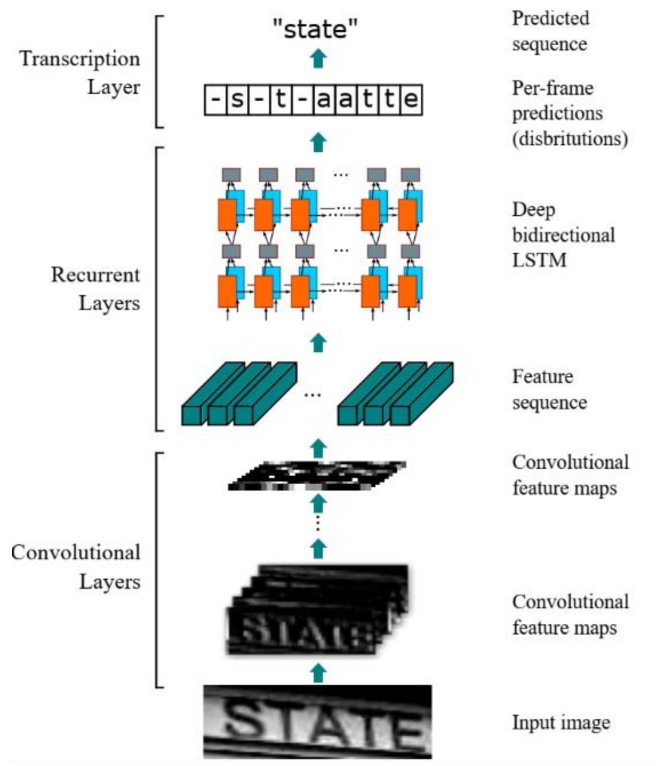
Concept

- WL(Weakly Labelled):Only know the **presence or absence** of sound events, without knowing their onset and offset times.

- AT(Audio Tagging): predict one or a few **labels** of an audio recording.
- SED(Sound Event Detection):predict the **onsets** and **offsets** of sound events.

Solution

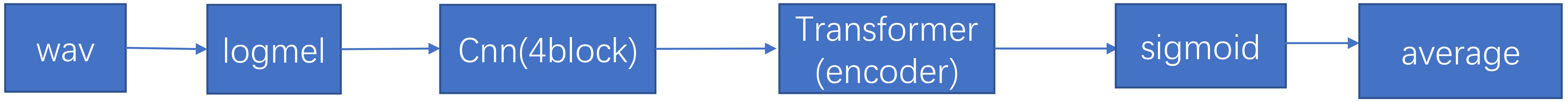
- CNNs:CNNs do not capture the long time dependency in an audio clip well
- CRNNs:hidden states of a CRNN have to be calculated one by one
- CNN-Transformer: each state retains the global information of the input sequence



crnn

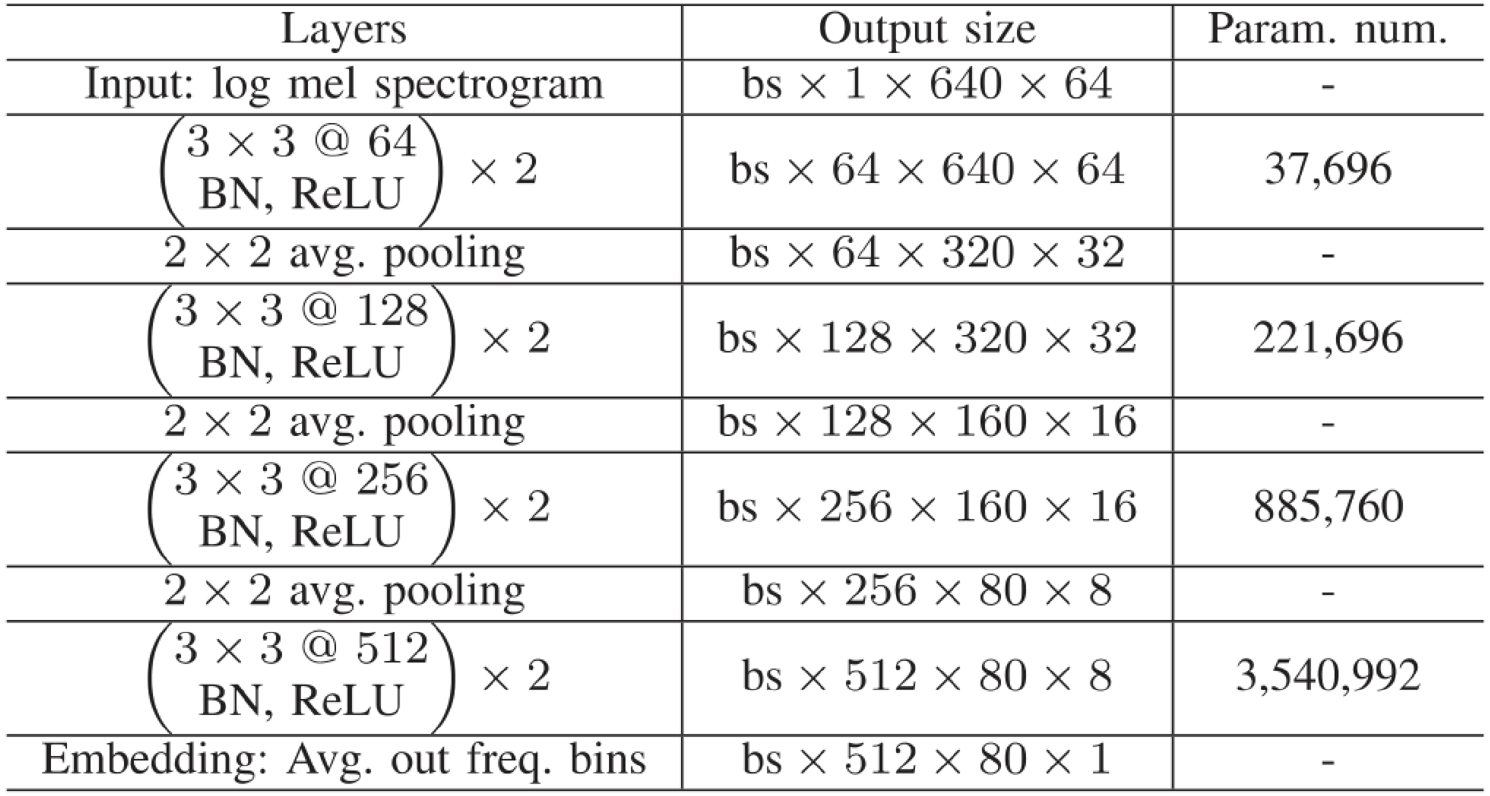
CNN-Transformer

- CNN: **High level features** are those extracted from low level features .
- Transformer :applies a **self-attention mechanism** which directly models relationships between all time steps in a sequence(**capture the correlation of speeches**).



CNN-Transformer

- CNN: High level features are those extracted from low level features .



Layers	Output size	Param. num.
Input: log mel spectrogram	bs × 1 × 640 × 64	-
$\begin{pmatrix} 3 \times 3 \ @ \ 64 \\ \text{BN, ReLU} \end{pmatrix} \times 2$	bs × 64 × 640 × 64	37,696
2 × 2 avg. pooling	bs × 64 × 320 × 32	-
$\begin{pmatrix} 3 \times 3 \ @ \ 128 \\ \text{BN, ReLU} \end{pmatrix} \times 2$	bs × 128 × 320 × 32	221,696
2 × 2 avg. pooling	bs × 128 × 160 × 16	-
$\begin{pmatrix} 3 \times 3 \ @ \ 256 \\ \text{BN, ReLU} \end{pmatrix} \times 2$	bs × 256 × 160 × 16	885,760
2 × 2 avg. pooling	bs × 256 × 80 × 8	-
$\begin{pmatrix} 3 \times 3 \ @ \ 512 \\ \text{BN, ReLU} \end{pmatrix} \times 2$	bs × 512 × 80 × 8	3,540,992
Embedding: Avg. out freq. bins	bs × 512 × 80 × 1	-

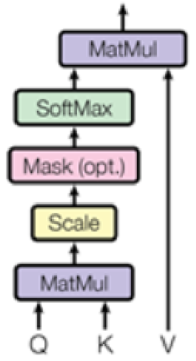
CNN-Transformer

- Encoder:

  W^Q :query transform matrix $\quad Q = xW^Q$

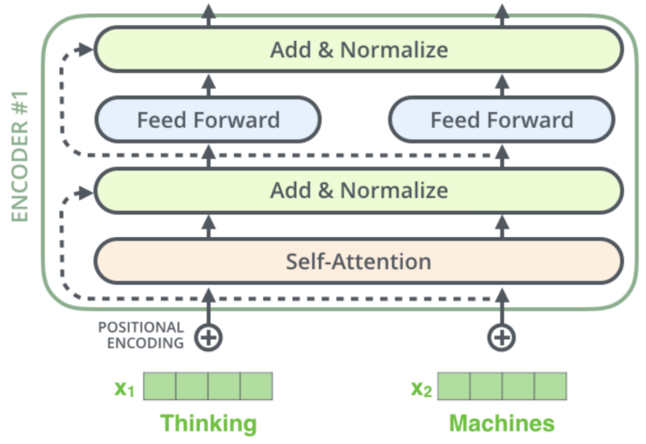
  W^K : key transform matrix $\quad K = xW^K$

  W^V :value transform matrix $\quad V = xW^V$



- The output of an encoder layer

$$h = soft\max(\frac{QK^T}{\sqrt{d_k}})V$$



CNN-Transformer

- Step 1： log-mel spectrogram
- Step 2： CNN
- Step 3： last convolutional layer to obtain embedding vectors
- Step 4： transformer
- Step 5： sigmoid non-linearity
- Step 6： average probabilities

Segment-wise vs Clip-wise

- Segment-wise Training

  Each segment inherits the tags of the audio clip.
  The SED problem is converted to an audio tagging problem on those segments.

$$E=-\sum_{m=1}^{M}\sum_{k=1}^{K}[y_k \log f(x_m)_k + (1-y_k)\log(1-f(x_m)_k)]$$

X : an audio clip

M :the number of segments

x_m :split X

$y \in \{0,1\}^K$:labels

f :classifier

Segment-wise vs Clip-wise

- Segment-wise Training

  AT result :aggregating $f(x_m)$

  $$F(X) = agg(\{f(x_m)\}_{m=1}^{M})$$

  Aggregation: maximum , average

Segment-wise vs Clip-wise

- Clip-wise Training
  - Idea:does not explicitly assign tags for each segment x_m, learn the tags of x_m

$$F(X)_k = \sum_{m=1}^{M} f(x_m)_k \, p(x_m)_k$$

  - Where $p(x_m) = \frac{\exp(w(x_m)_k)}{\sum_{j=1}^{M} \exp(w(x_j)_k)}$, w(·) a linear transformation

$$E = -\sum_{k=1}^{K} [y_k \log F(X)_k + (1 - y_k) \log(1 - F(X)_k)]$$

Post-processing

- Thresholds: In the AT subtask, if the predicted probability of a sound class is over a threshold in an audio clip, then the audio clip is regarded as containing this sound class.

- The selection of thresholds: empirically,Automatic threshold optimization

Automatic threshold optimization

- Threshold :To obtain the presence or absence of sound events in an audio clip, AT systems need to apply thresholds to the system outputs.

$$\Theta^{AT} = \{\mu_1, ..., \mu_K\}$$

$$\Theta^{SED} = \{\mu_1, ..., \mu_K, \tau_1^{high}, ..., \tau_K^{high}, \tau_1^{low}, ..., \tau_K^{low}\}$$

Automatic threshold optimization

- calculate the gradients over the thresholds in a numerical way

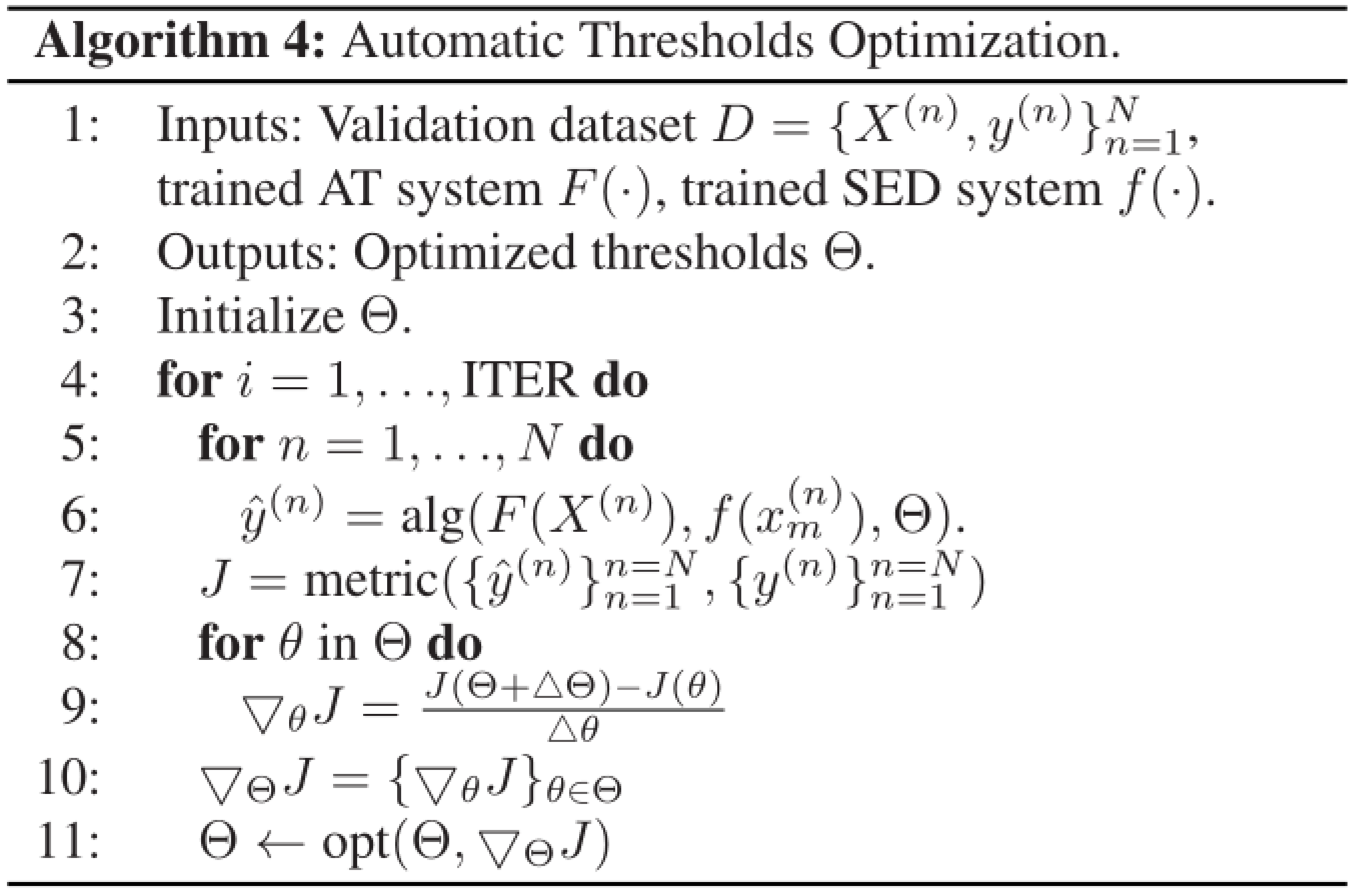
$$\hat{\Theta} = \arg\min_{\Theta} J(\Theta)$$

$$\nabla_{\theta} J(\Theta) = \frac{J(\Theta + \Delta\Theta) - J(\Theta)}{\Delta\theta}$$

$\Delta\theta$:a small constant number

$\Delta\Theta$: is a vector with all zero values the position of θ

Automatic threshold optimization



Algorithm 4: Automatic Thresholds Optimization.

```
1:  Inputs: Validation dataset $D = \{X^{(n)}, y^{(n)}\}_{n=1}^{N}$,
    trained AT system $F(\cdot)$, trained SED system $f(\cdot)$.
2:  Outputs: Optimized thresholds $\Theta$.
3:  Initialize $\Theta$.
4:  for $i = 1, \ldots,$ ITER do
5:    for $n = 1, \ldots, N$ do
6:      $\hat{y}^{(n)} = \mathrm{alg}(F(X^{(n)}), f(x_m^{(n)}), \Theta)$.
7:    $J = \mathrm{metric}(\{\hat{y}^{(n)}\}_{n=1}^{n=N}, \{y^{(n)}\}_{n=1}^{n=N})$
8:    for $\theta$ in $\Theta$ do
9:      $\bigtriangledown_{\theta} J = \frac{J(\Theta + \triangle\Theta) - J(\theta)}{\triangle\theta}$
10:   $\bigtriangledown_{\Theta} J = \{\bigtriangledown_{\theta} J\}_{\theta \in \Theta}$
11:   $\Theta \leftarrow \mathrm{opt}(\Theta, \bigtriangledown_{\Theta} J)$
```

Metrics

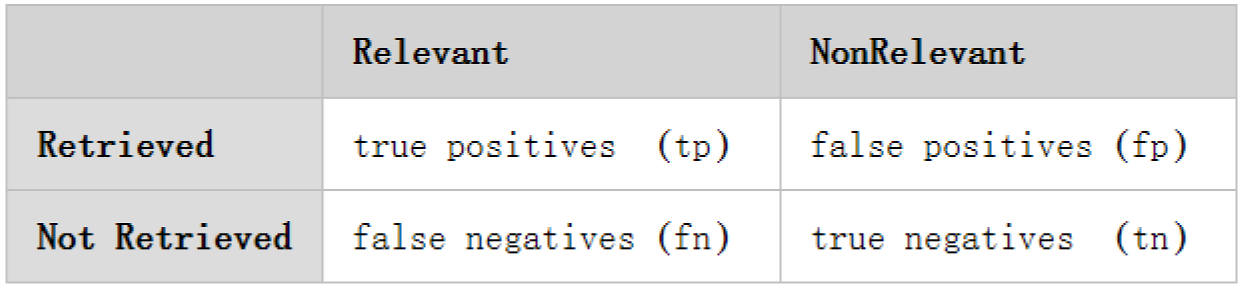
- Evaluation Metrics

$$F1 = \frac{2P \cdot R}{P + R}$$

$$ER = \frac{\sum_m S(m) + \sum_m D(m) + \sum_m I(m)}{\sum_m N(m)}$$

$$P = \frac{TP}{TP + FP}$$

$$R = \frac{TP}{TP + FN}$$



	Relevant	NonRelevant
Retrieved	true positives (tp)	false positives (fp)
Not Retrieved	false negatives (fn)	true negatives (tn)

准确率就是找得对，召回率就是找得全

Dataset

	Event name	**Training number**
Warning sounds	Train horn	441
	Air horn, trunk horn	407
	Car alarm	273
	Reversing beeps	337
	Ambulance (siren)	624
	Police car (siren)	2399
	Fire engine, fire trunk(siren)	2399
	Civil defense siren	1506
	Screaming	744
Vehicle sounds	Bicycle	2020
	Skateboard	1617
	Car	25744
	Car passing by	3724
	Bus	3745
	Trunk	7090
	Motorcycle	3291
	Train	2301
Gunshot	Gunshot	606

Experiment

- Recurrence
  - AT

Model	F1	Precision	Recall
CNN-biGRU-Att	0.598	0.614	**0.584**
CNN-Transformer-Ave	**0.617**	**0.661**	0.579

  - SED

Model	F1	Error rate
CNN-biGRU-Att	0.509	**0.683**
CNN-Transformer-Ave	**0.516**	0.719

Experiment（18class）

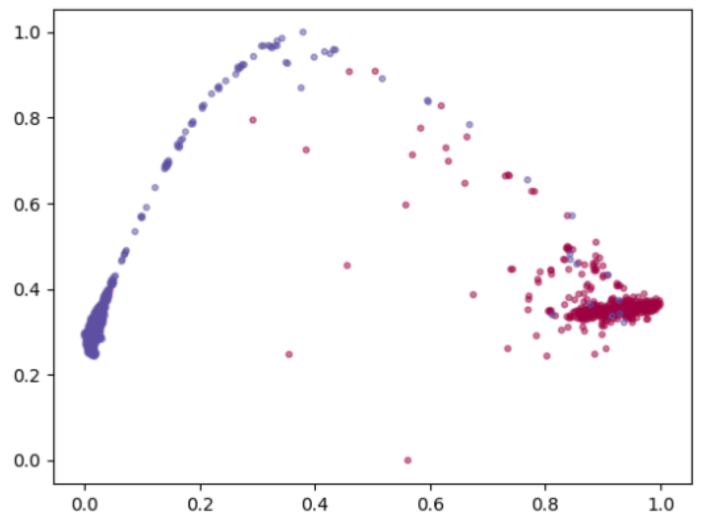
- AT

Model	F1	Precision	Recall
CNN-biGRU-Att	0.650	0.686	0.617
CNN-Transformer-Ave	0.626	0.647	0.606

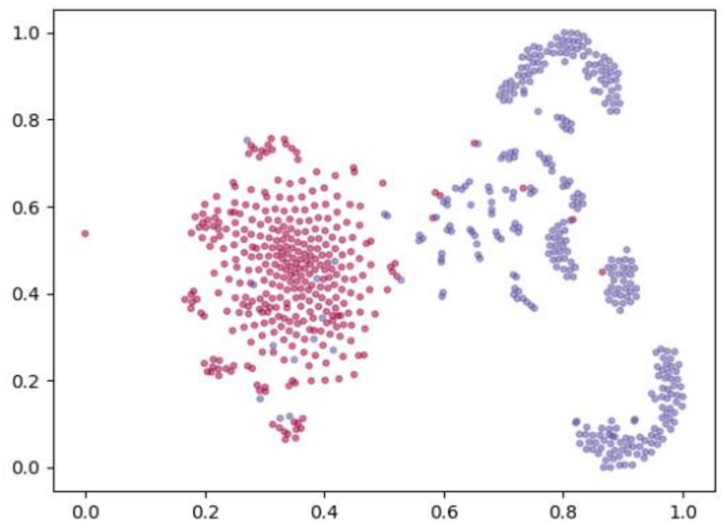
- SED

Model	F1	Error rate
CNN-BIGRU-Att	0.582	0.615
CNN-Transformer-Ave	0.531	0.741

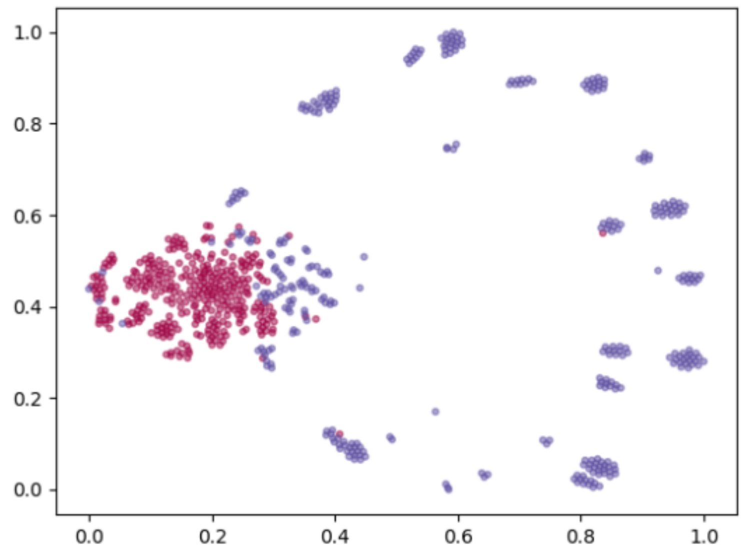
Experiment



Original space

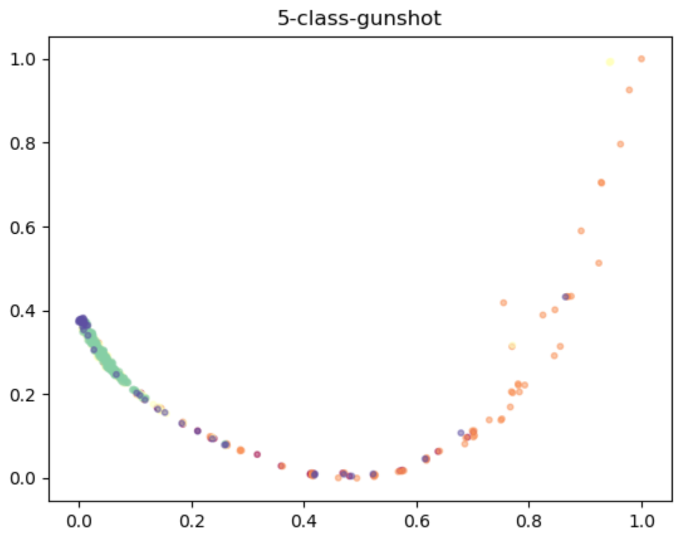


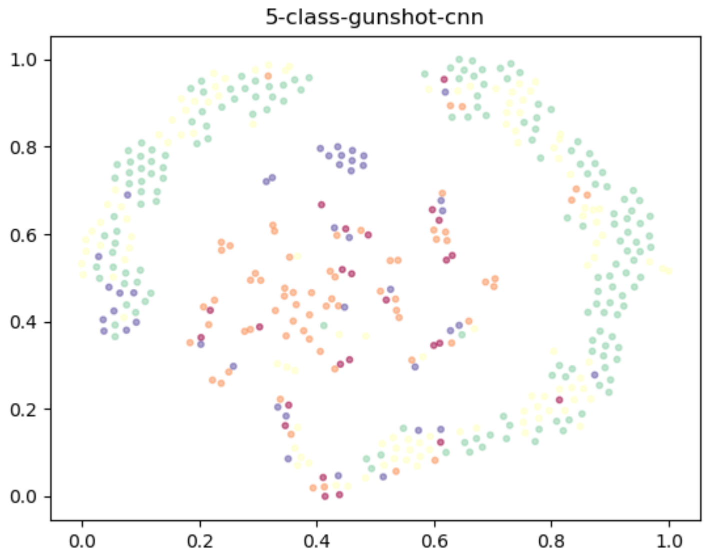
Latent space （cnn）

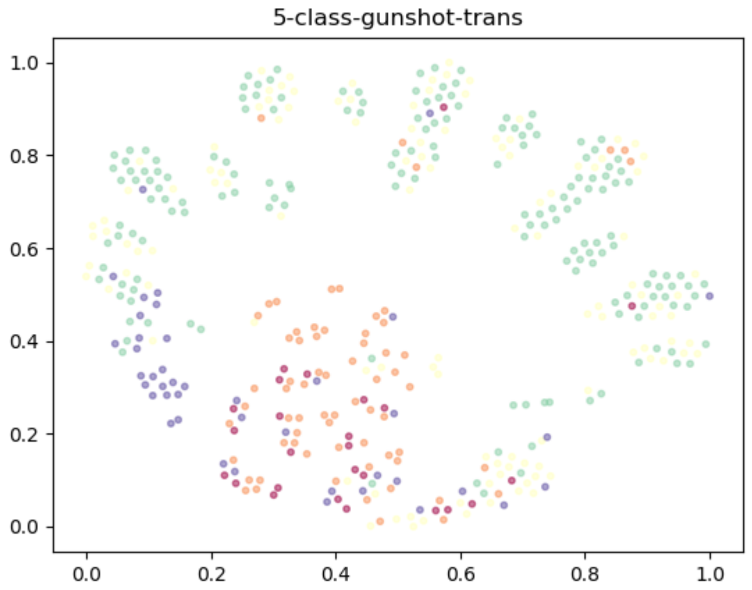


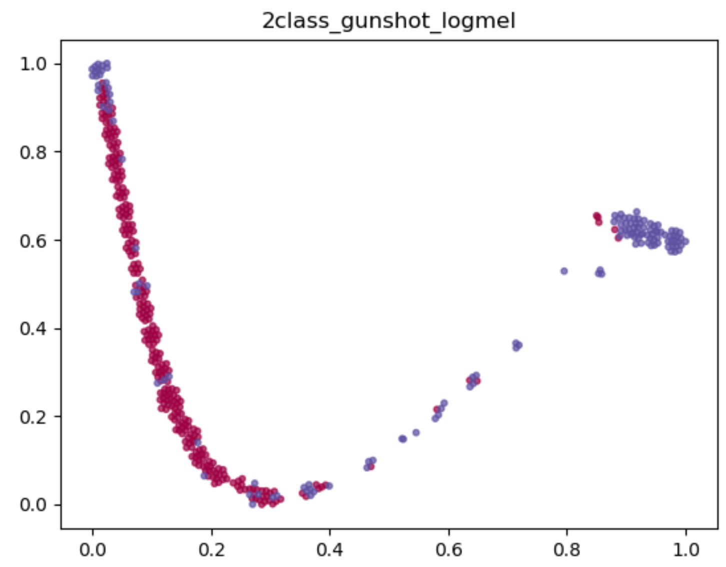
Latent space(cnn-transformer)

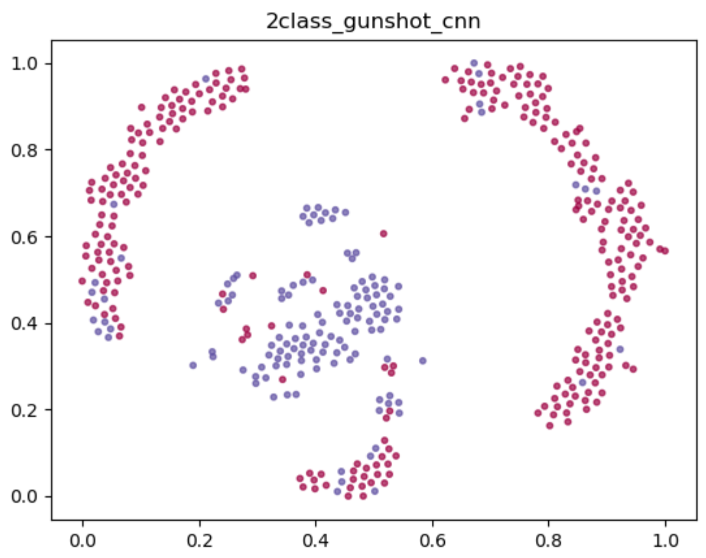
Experiment

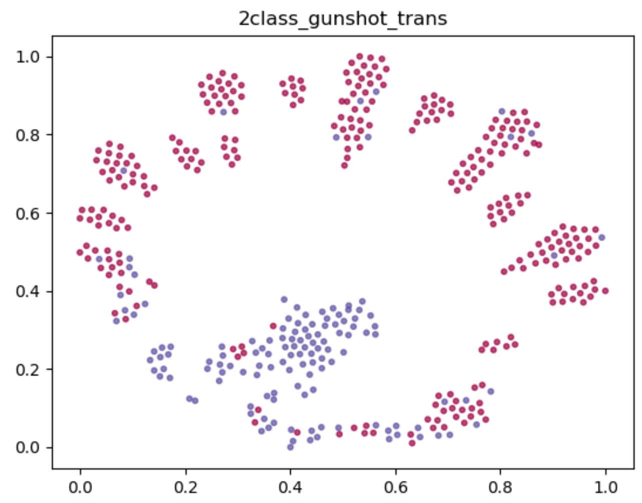

5-class-gunshot
0.0
0.2
0.4
0.6
0.8
1.0

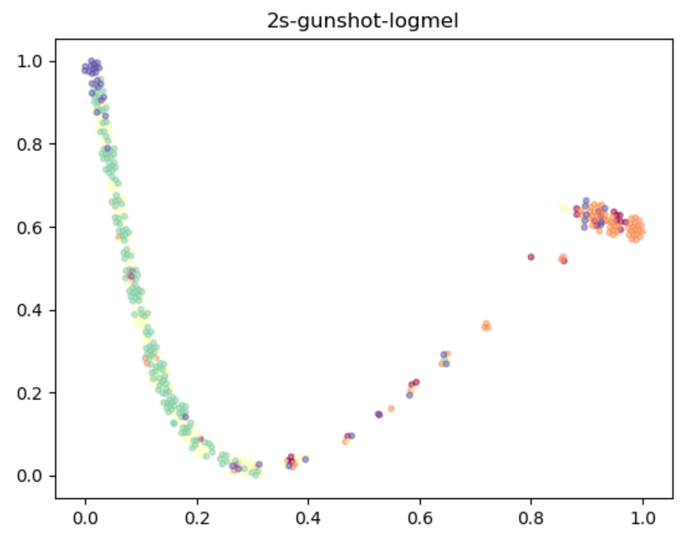

5-class-gunshot-cnn
0.0
0.2
0.4
0.6
0.8
1.0

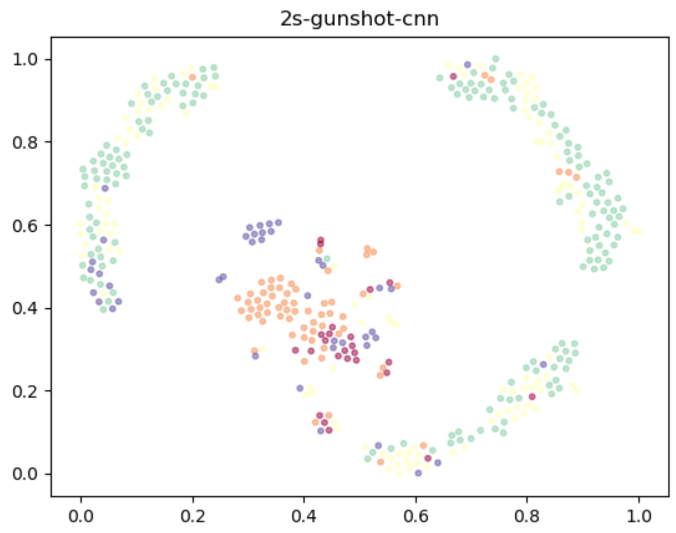

5-class-gunshot-trans
0.0
0.2
0.4
0.6
0.8
1.0

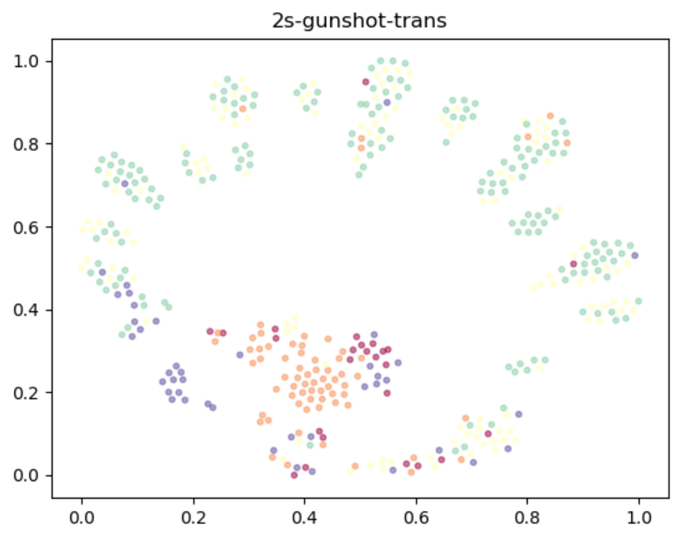






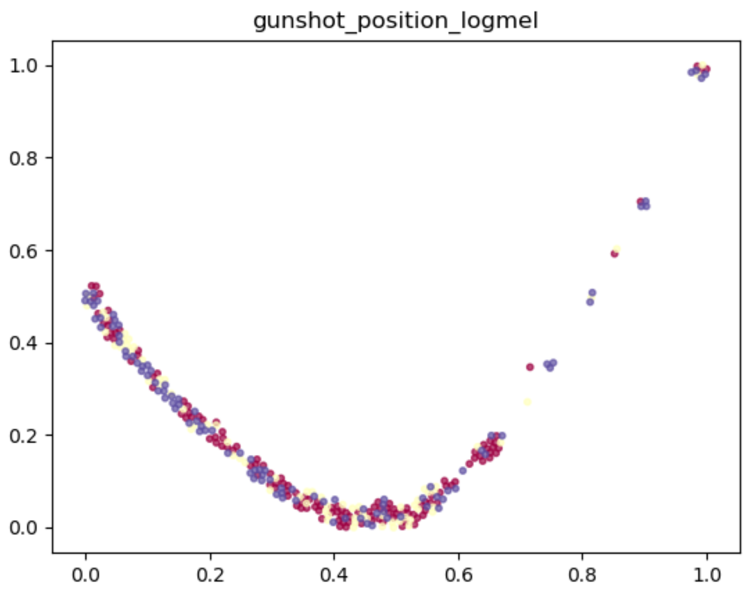




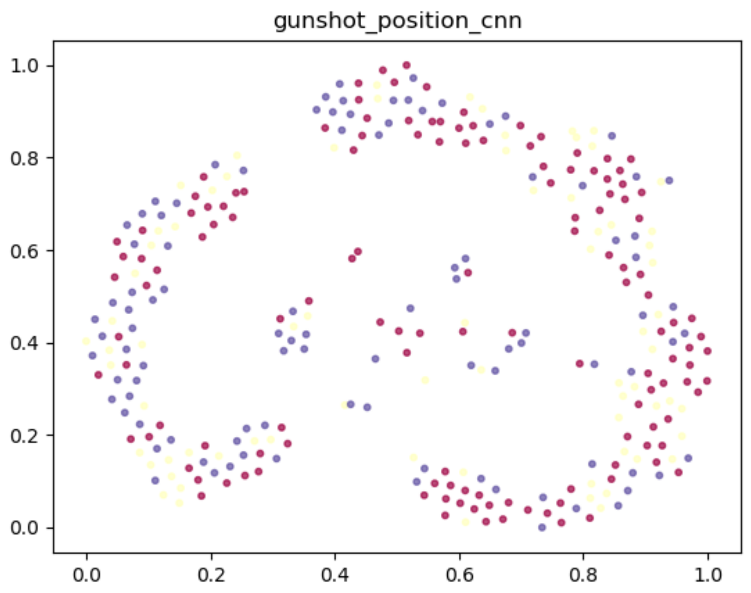


Green:riflee
Yellow:pistol
Orange:machinegun
Purple:submachinegun
Pink:assaultrifle

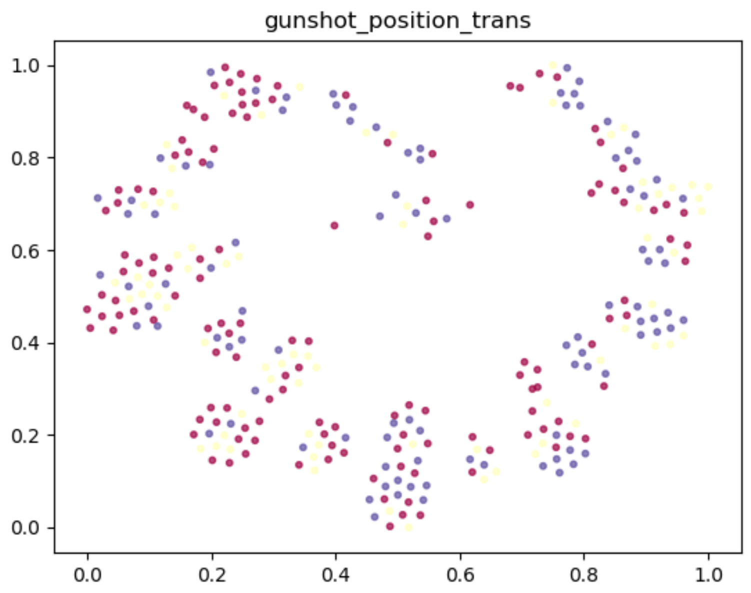
Experiment



gunshot_position_logmel
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0



gunshot_position_cnn
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0



gunshot_position_trans
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

Experiment

Gun&nogun	训练数据	测试数据	标签
Gun	306	34	Gunshot
nogun	306	34	Other class

Gunshot class		训练数据	测试数据	标签
Rif&pis	Riflee	150	17	Rif_pis
	Pistol	100	15	
Mac&sub_ass	Machinegun	60	7	Mac_sub_ass
	Submachinegun	36	6	
	Assaultrifle	21	4	

position	训练数据	测试数据	标签
b	107	27	b
f	58	14	f
s	80	18	s

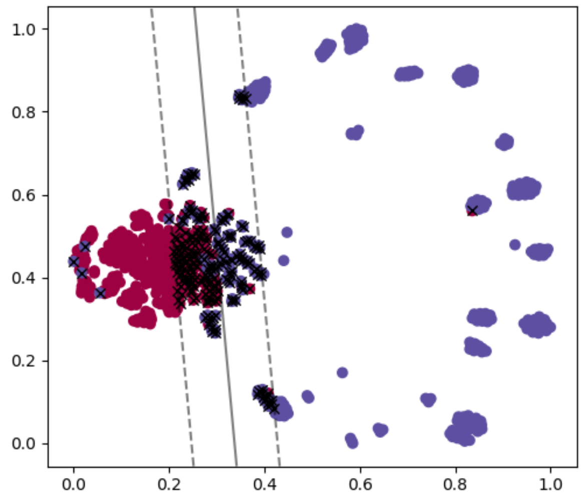
Experiment（18）

-		cnn_trans	cnn_trans+svm	logmel+svm	cnn_trans(17)+svm
gun & nogun	F1	0.975	0.987	0.971	0.957
	Precision	0.995	1	1	0.982
	Recall	0.956	0.974	0.943	0.933
	Accuracy	-	0.985	0.971	0.951
pistol & machinegun	F1	-	0.857	0.667	0.818
	Precision	-	1	0.6	0.9
	Recall	-	0.75	0.75	0.75
	Accuracy	-	0.929	0.786	0.905
position	F1(macro)	-	0.379	0.620	0.456
	Precision(macro)	-	0.380	0.638	0.480
	Recall(macro)	-	0.404	0.622	0.498
	Accuracy	-	0.419	0.613	0.516

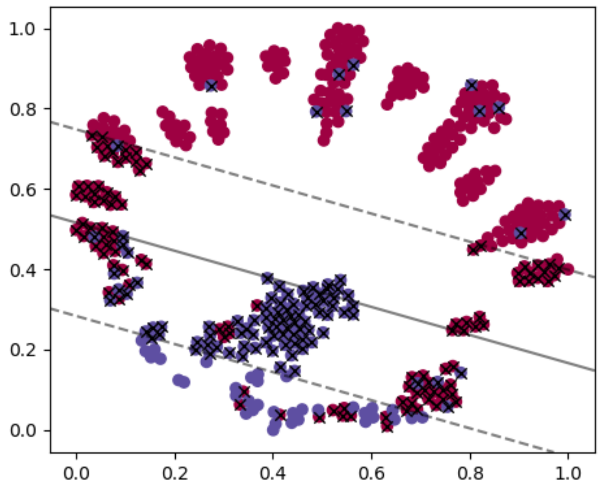
Experiment

position		Cnn_trans+svm	Logmel+svm	Cnn_trans(17)+svm	Cnn_trans(modify trans)+svm	Cnn_trans(modify cnn+trans)+svm
	F1	0.543	0.521	0.536	0.521	0.586
	Precision	0.547	0.533	0.559	0.521	0.591
	Recall	0.545	0.519	0.537	0.524	0.587
	Accuracy	0.557	0.541	0.574	0.557	0.607

Cnn_trans+svm

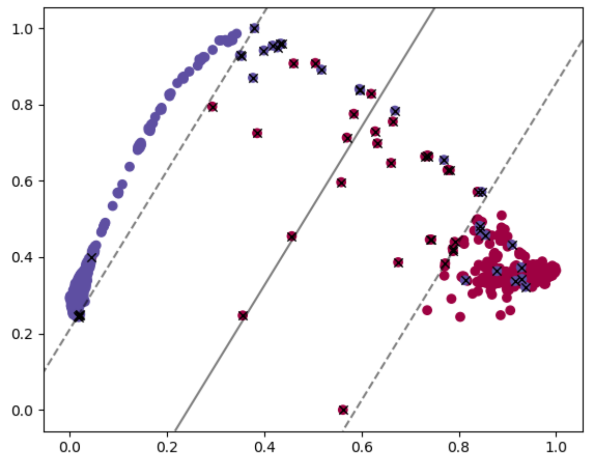


Gun&nogun

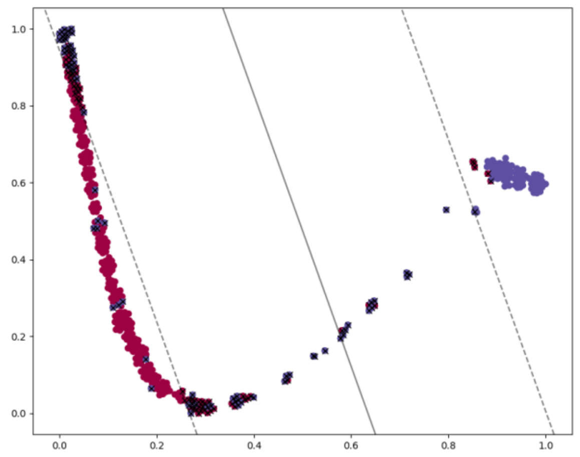


pistol & machinegun

logmel+svm

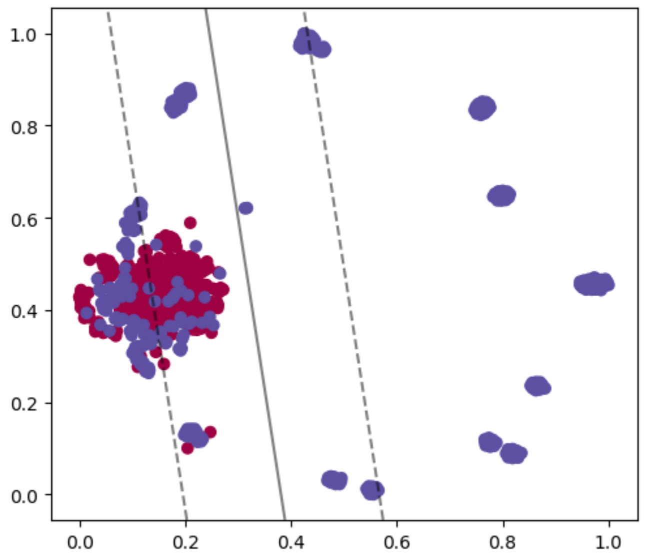


Gun&nogun

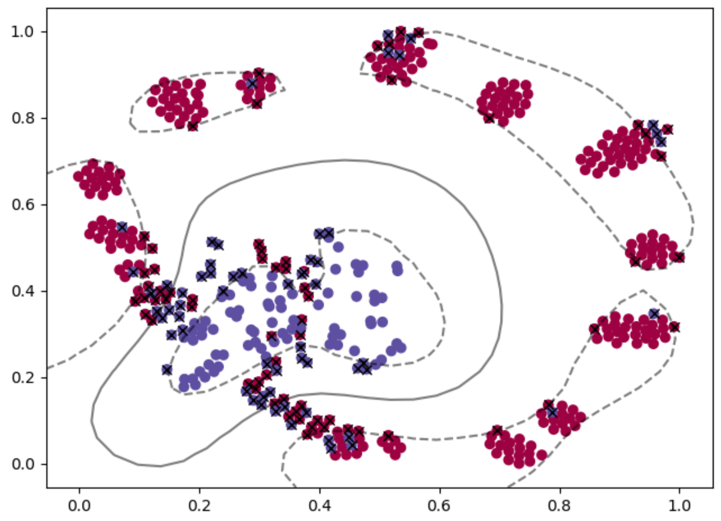


pistol & machinegun

Cnn_trans(17)+svm



Gun&nogun



pistol & machinegun

Summary

- Cnn-transformer has a lot of room for improvement.
- Continue to expand the amount of gunshot data.

Thank you!