Weakly-supervised deep learning approaches for sound event detection

Thomas Pellegrini, Léo Cances

GDR ISIS, 10-05-2019, Paris

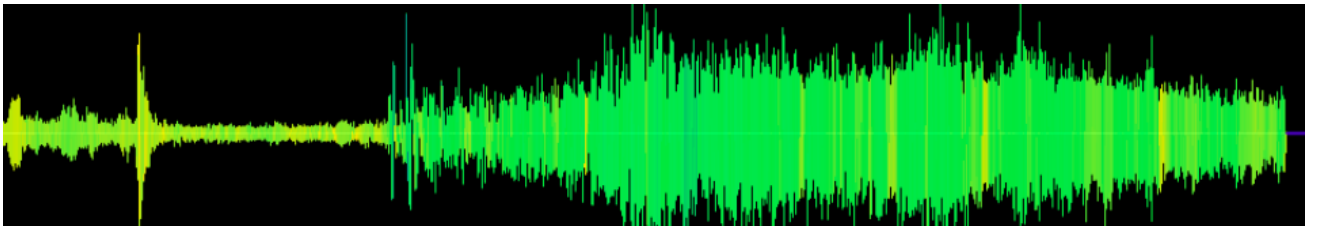
Outline

LUDAU

ANR JCJC Project (ANR-18-CE23-0005-01)

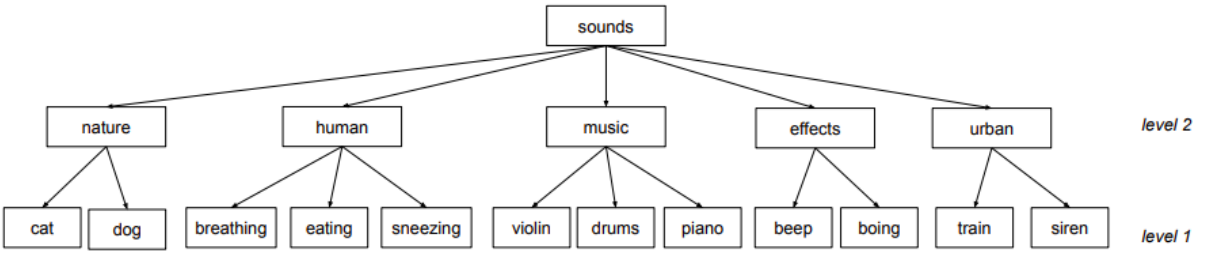
- Lightly-supervised and Unsupervised Discovery of Audio Units using Deep Learning
- Start: 01 Oct. 2018
- End: 31 March 2022 (duration: 42 months)
- PhD student: Léo Cances, Post-doc to be hired

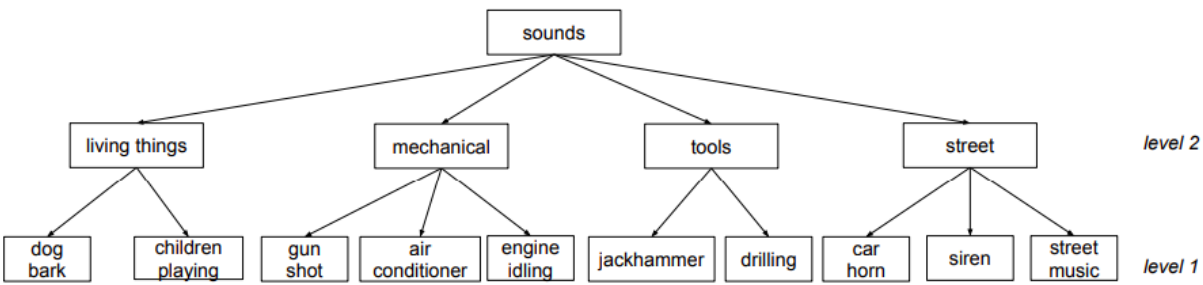
Machine listening



... extracting information from sound

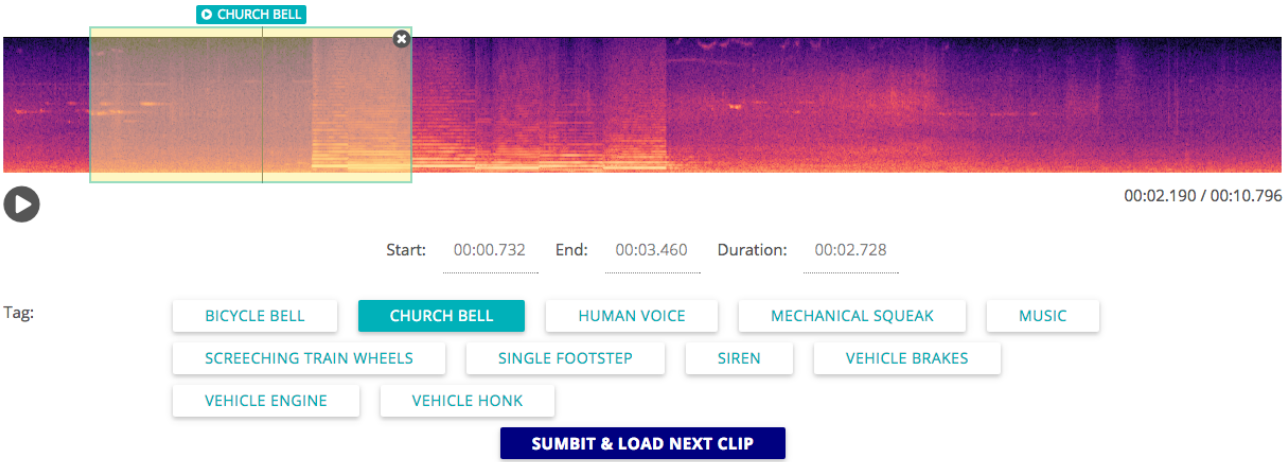
Machine listening





... making sense of sound

Why weakly-supervised learning?



`https://github.com/CrowdCurio/audio-annotator/`

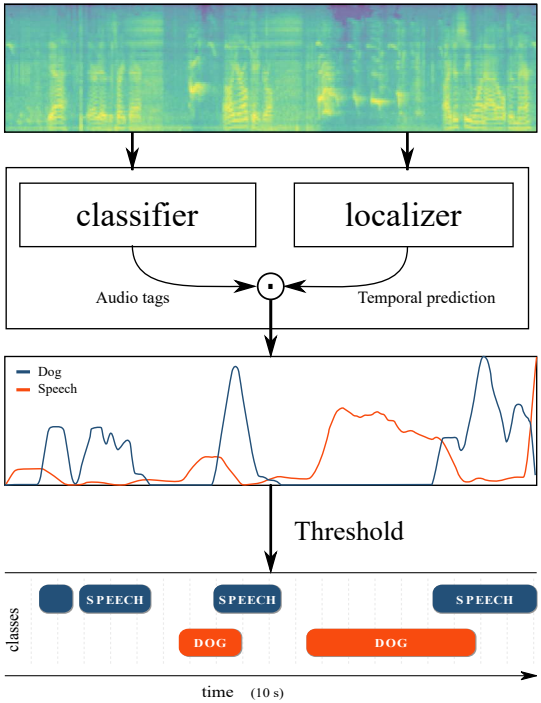
M. Cartwright, A. Seals, J. Salamon, A. Williams, S. Mikloska, D. MacConnell, E. Law, J. Bello, and O. Nov. "Seeing sound: Investigating the effects of visualizations and complexity on crowdsourced audio annotations." In Proceedings of the ACM on Human-Computer Interaction, 1(1), 2017.

Why weakly-supervised learning?

Issues regarding data annotation

- What to annotate? Which granularity?
- Annotations are limited, expensive and time-consuming
- Annotations may introduce biases
- Annotations are noisy, "label noise"
- "Digital labor", les travailleurs du clic, Antonio Casilli

Sound Event Detection



- Multi-label classification problem
- *Weak labels*: audio tags, $\mathbf{y} \in \{0,1\}^C$: $\rightarrow$ many-to-one RNN
- *Strong labels*: frame-level tags, $Y \in \{0,1\}^{T \times C}$: $\rightarrow$ many-to-many RNN

Weakly-labeled SED: problem statement

- We want to perform:
  - "audio tagging": infer *weak* labels, $\hat{\mathbf{y}}$
  - "localization" or SED: infer *strong* labels, $\hat{Y}$
- Input: a sequence of T.F. representation vectors, $X = \{\mathbf{x_1}, \ldots \mathbf{x_T}\}$
- **Only weak labels are available for learning**
- In a recording weakly labeled as *Dog*, some acoustic frames will comprise dog barking and others will not

Weakly-labeled SED: approaches

Three main approaches:

1. Baseline brute-force approach: "False strong labeling"
2. Attention mechanisms: Gated Linear Units, for ex.
3. **MIL-inspired approach: max MIL pooling function and variants**

Multiple Instance Learning

- MIL terminology
  - $X = \{\mathbf{x_1}, \ldots, \mathbf{x_T}\}$: a bag or a set
  - $\mathbf{x_i}$: an instance
  - No dependency nor ordering among instances
  - A single binary label Y
  - Unknown individual labels: $y = \{y_1, \ldots, y_T\}$

MIL assumptions:

$$Y = \begin{cases} 0, & \text{iff } \sum_t y_t = 0, \\ 1, & \text{otherwise} \end{cases}$$

In a compact form:

$$Y = \max_t \{y_t\}$$

MIL: how to model the bag-level prob $P(Y = 1|X)$?

- MIL assumption: $P(Y = 1|X)$ must be **permutation invariant** since no ordering nor dependency between instances $\mathbf{x_t}$
- Instances: "monomials"

Theorem 1. *A scoring function for a set of instances X, $S(X) \in \mathbb{R}$, is a symmetric function (i.e., permutation-invariant to the elements in X), if and only if it can be decomposed in the following form:*

$$S(X) = g\Big(\sum_{\mathbf{x} \in X} f(\mathbf{x})\Big), \tag{3}$$

where f and g are suitable transformations.

Theorem 2. *For any $\varepsilon > 0$, a Hausdorff continuous symmetric function $S(X) \in \mathbb{R}$ can be arbitrarily approximated by a function in the form $g\big(\max_{\mathbf{x} \in X} f(\mathbf{x})\big)$, where $\max$ is the element-wise vector maximum operator and f and g are continuous functions, that is:*

$$|S(X) - g\big(\max_{\mathbf{x} \in X} f(\mathbf{x})\big)| < \varepsilon. \tag{4}$$

MIL: how to model $P(Y = 1|X)$?

Instance-level approach

- f: instance-level classifier that returns scores for each instance
- σ: aggregates the scores
- g: Identity function

Bag-level (embedding-level) approach

- f: maps instances to embeddings
- σ: maps the emb. to a single bag emb.
- g: bag-level classifier
- σ: "MIL pooling", basic non-learnable choices:
  - $\mathbf{z} = \max_{t=1\ldots T}\{\mathbf{h_t}\}$
  - $\mathbf{z} = 1/T \sum_{t=1}^{T}\{\mathbf{h_t}\}$

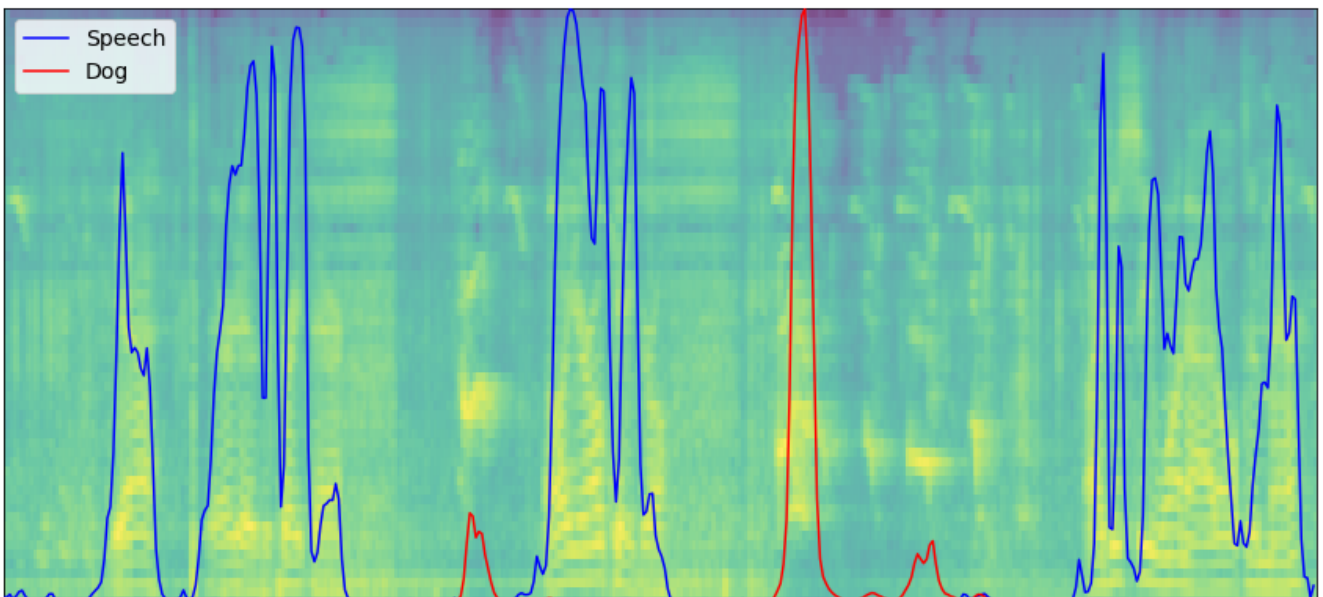
MIL loss example

$$\mathrm{loss}(\{X, Y_c\}) = \mathrm{binCE}(Y_c, \max_t \hat{\mathbf{y}}_{\mathbf{tc}})$$

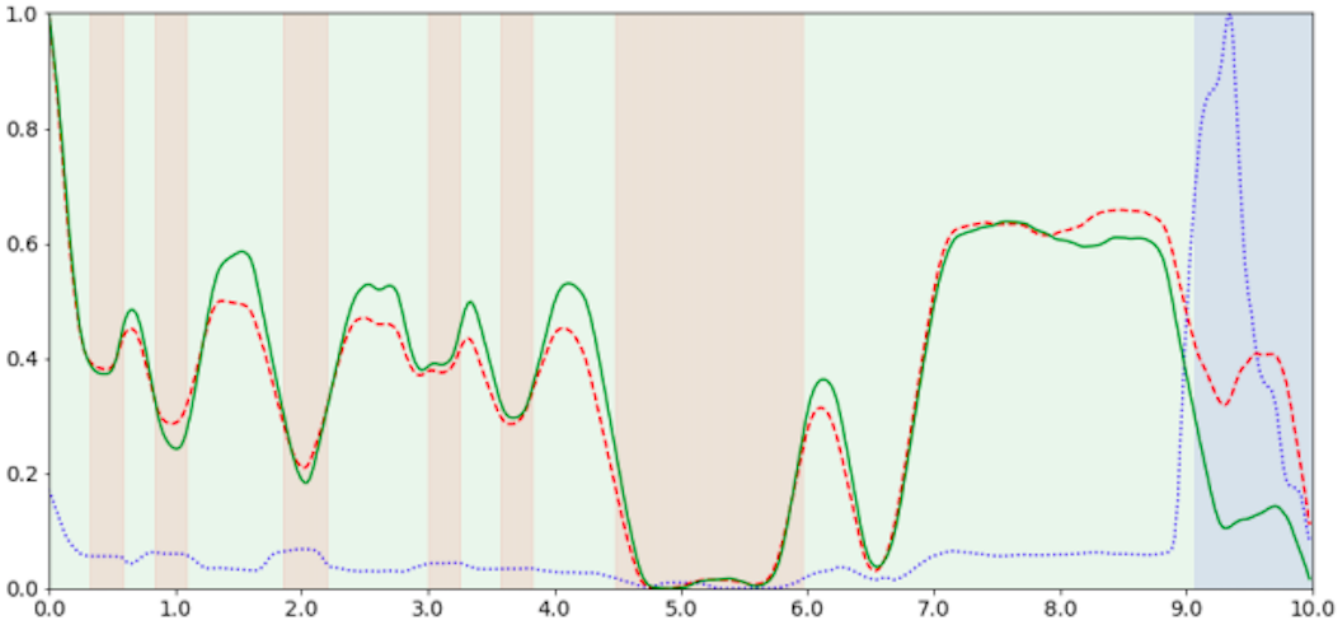
where

- $Y_c \in \{0, 1\}$: ground truth label for class c
- $\hat{\mathbf{y}}_{\mathbf{tc}} \in [0, 1]^T$: temporal predictions for class c

Example OK

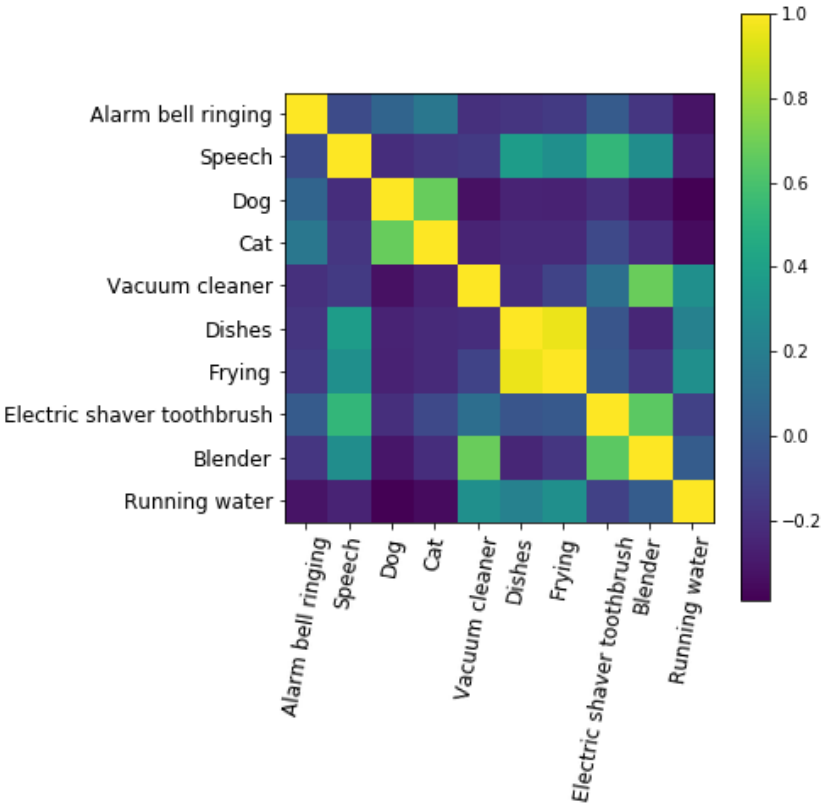


Example KO



- Issue in distinguishing some classes

Prediction correlations



- Confusions between *Dog* and *Cat*, why not
- Confusions between *Dishes* and *Frying*, why??

Prediction correlations

- Confusions between *Dog* and *Cat*, why not
- Confusions between *Dishes* and *Frying* $\rightarrow$ score $= 0$ for *Dishes*, why??

$\rightarrow$ A caveat of the MIL pooling function (here max)

- When a class most often co-occurs with another one
- When max predictions are scarce (a single frame possibly): during training the gradient flows on those frames only

Prediction correlations

- Confusions between *Dog* and *Cat*, why not
- Confusions between *Dishes* and *Frying* $\rightarrow$ score $= 0$ for *Dishes*, why??

$\rightarrow$ A caveat of the MIL pooling function (here max)

- When a class most often co-occurs with another one
- When max predictions are scarce (a single frame possibly): during training the gradient flows on those frames only

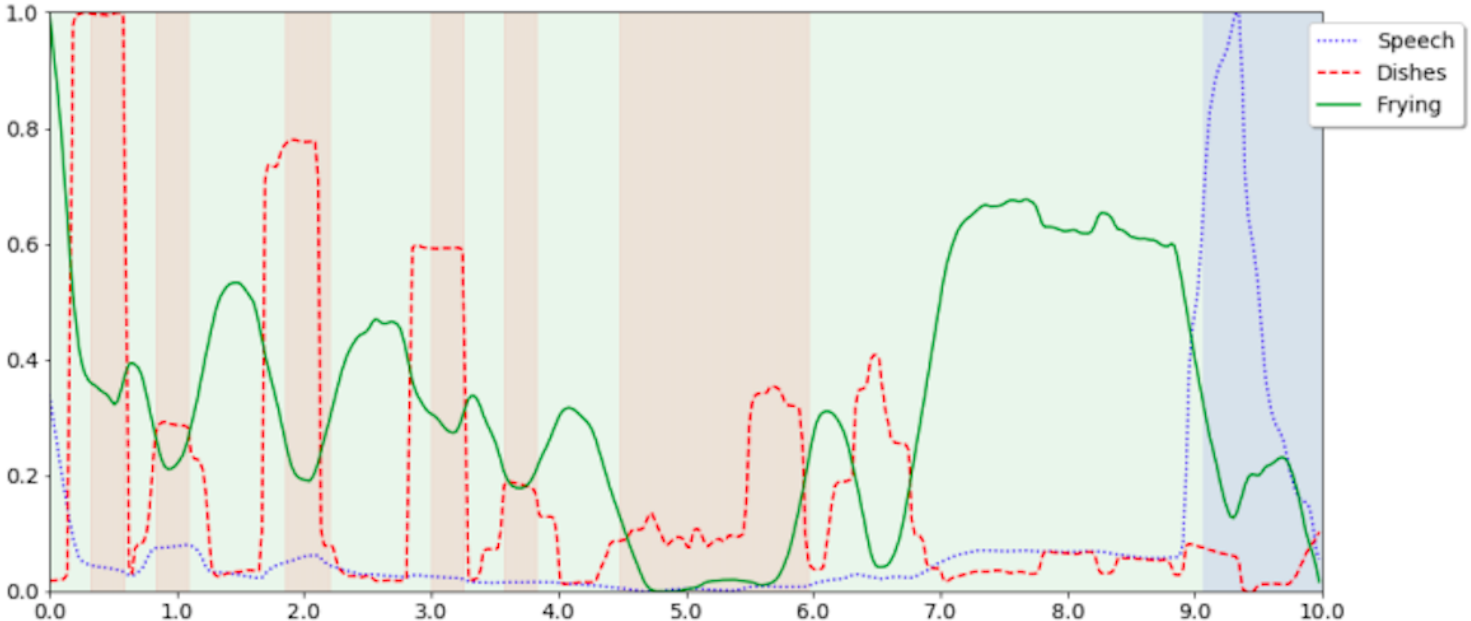
Solutions?

- Choose another MIL pooling function (ongoing work)?
- Data augmentation (ongoing work)?
- **Penalize for prediction similarity**

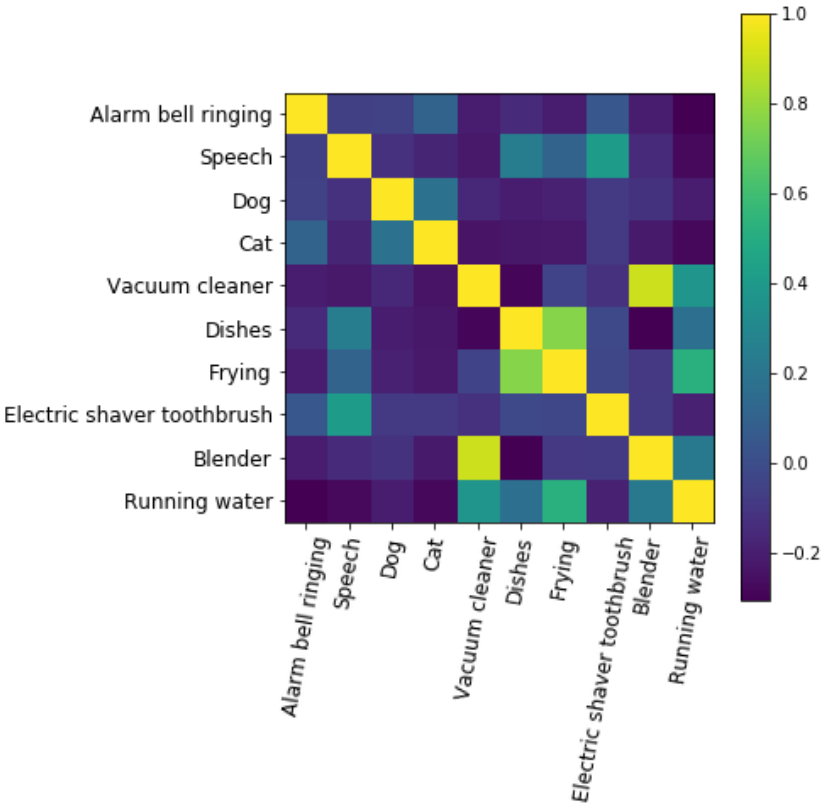
Adding a similarity penalty

$$\begin{aligned}\mathrm{loss}(\{X, Y_c\}) &= \mathrm{binCE}(Y_c, \max_t \hat{\mathbf{y}}_{\mathbf{tc}}) \\ &\quad + \alpha\, Y_c \sum_{l \neq c}^{C} Y_l \max(0, \cos(\hat{\mathbf{y}}_{:\mathbf{l}}, \hat{\mathbf{y}}_{:\mathbf{c}}))\end{aligned}$$

Adding a similarity penalty

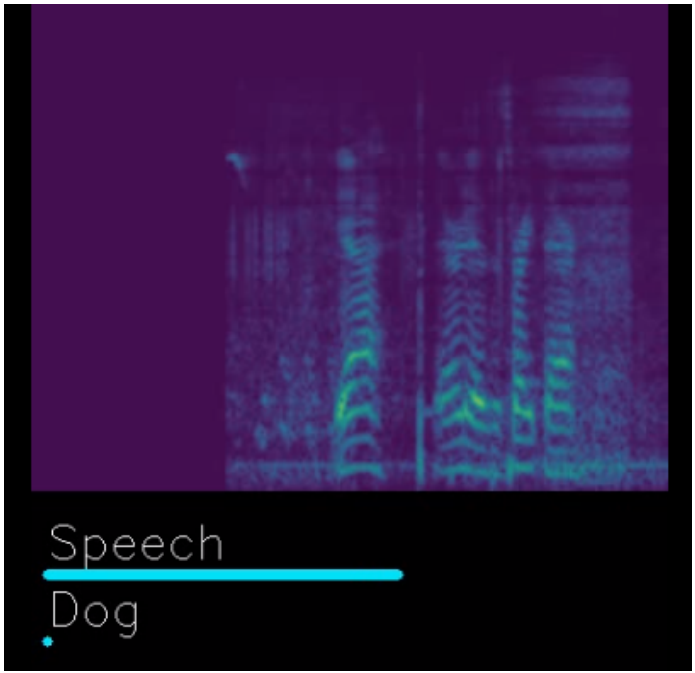


Prediction correlations



- Less confusions!

demo



Results on DCASE 2018 Eval subset

Approach	Official Baseline	GLU-MIL	GLU-MIL+cos	JiaKai	Liu
F-score (%)	10.80	22.60	26.20	32.4	29.9
Alarm / bell / ringing	4.8	29.0	30.4	49.9	46.0
Blender	12.7	23.3	27.7	38.2	27.1
Cat	2.9	28.7	30.3	3.6	20.3
Dishes	0.4	0.0	19.0	3.2	13.0
Dog	2.4	19.8	20.9	18.1	26.5
Electric shaver / toothbrush	20.0	6.2	19.1	48.7	37.6
Frying	24.5	27.6	21.2	35.4	10.9
Running water	10.1	13.3	13.2	31.2	23.9
Speech	0.1	37.6	35.0	46.8	43.1
Vacuum cleaner	30.2	40.6	45.2	48.3	50.0

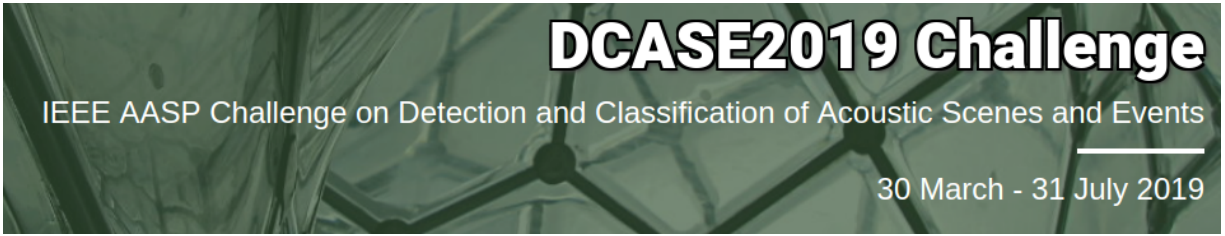
Discussion on the cos penalty

Pros

- Increased the class-discriminative power of the networks

Open questions

- Larger variance
- Tends to force events to not overlap (but OK for the challenge)
- Better to penalize the prediction distributions? e.g. a Kullback-Leibler penalty?
- Worked for SED, generalization to other MIL problems?



Tasks

- Acoustic Scene Classification
- Audio tagging with noisy labels and minimal supervision
- Sound Event Localization and Detection
- Sound event detection in domestic environments
- Urban Sound Tagging

Task 4 focus:

- do we really need real but partially and weakly annotated data or is using simulated data sufficient? or do we need both?

References

- T. Pellegrini, L. Cances, Cosine-similarity penalty to discriminate sound classes in weakly-supervised sound event detection, to appear in Proc. IJCNN, July 2019, Budapest, arxiv.org/abs/1901.03146
- L. Cances, T. Pellegrini, P. Guyot, Sound event detection from weak annotations: weighted-GRU versus multi-instance-learning, In Proc. IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events (DCASE 2018), pp. 64-68, Surrey, Nov. 2018