Can Self-Supervised Neural Representations Pre-Trained on Human Speech distinguish Animal Callers?

Eklavya Sarkar\textsuperscript{1,2}, Mathew Magimai Doss\textsuperscript{2}

\textsuperscript{1} Idiap Research Institute, Switzerland
\textsuperscript{2} Ecole polytechnique fédérale de Lausanne, Switzerland

ISCA Interspeech 2023
August 2023
Contents

- Introduction
- Study Design
- Caller Discrimination Analysis
- Caller Detection Study
- Summary and Open Questions
1. Introduction
Bio-Acoustics (Animal Vocalizations)

Topic:
● Study of animal vocalizations.
● Research has progressed in recent due to approaches inherited from ML/DL.
Bio-Acoustics (Animal Vocalizations)

Topic:
- Study of animal vocalizations.
- Research has progressed in recent due to approaches inherited from ML/DL.

Issues:
- Labeled data scarcity.
- Lack of domain knowledge.
- Understudied topic.
Bio-Acoustics (Animal Vocalizations)

Topic:
- Study of animal vocalizations.
- Research has progressed in recent due to approaches inherited from ML/DL.

Issues:
- Labeled data scarcity.
- Lack of domain knowledge.
- Understudied topic.

- Self-supervised learning has emerged as a way of leveraging unlabeled data.
Self-Supervised Learning Framework

Pre-training:
- Create surrogate labels from unlabeled data based on the pre-text task.
- Optimize its learning objective.
Self-Supervised Learning Framework

Pre-training:
- Create surrogate labels from unlabeled data based on the pre-text task.
- Optimize its learning objective.
- Goal: learn useful representations.
Self-Supervised Learning Framework

Pre-training:
- Create surrogate labels from unlabeled data based on the pre-text task.
- Optimize its learning objective.
- Goal: learn useful representations.
- Network infers intrinsic structure.
- No knowledge is explicitly provided (e.g. speech production mechanism).
- Utility not limited to modeling speech.
Self-Supervised Pre-Training Objectives

- The information encoded in the SSL representations can vary depending on *learning objective* (among other elements).
Self-Supervised Pre-Training Objectives

- The information encoded in the SSL representations can vary depending on *learning objective* (among other elements).
- These can be roughly categorized into the four approaches given below.
- This framework has yielded SOTA results on the SUPERB benchmark.
Humans and animals share a commonality: they both have voice production mechanism.
Motivation

- Given this understanding, our objective is to:
  - Investigate the cross-transferability of pre-trained SSL representations learned from human speech for analyzing animal vocalizations.
Motivation

- Given this understanding, our objective is to:
  - Investigate the cross-transferability of pre-trained SSL representations learned from human speech for analyzing animal vocalizations.
- Previous works has explored birdsong detection\(^1\) and bio-acoustic event detection\(^2\) using contrastive learning.
- However, the generalization of SSL representations to animal vocalizations has largely remained unexplored.
- Aim: distinguish individual identities within the same species (caller detection).

\(^1\)Saeed et al., *Contrastive learning of general-purpose audio representations*, ICASSP, 2021.
Research Questions

We design studies for following research questions:
Research Questions

We design studies for following research questions:

1. How discriminative are the embedding spaces of SSL models pre-trained on human speech?
Research Questions

We design studies for following research questions:

1. How discriminative are the embedding spaces of SSL models pre-trained on human speech?

2. Can we systematically detect individual animal callers using said embedding spaces?
Research Questions

We design studies for following research questions:

1. How discriminative are the embedding spaces of SSL models pre-trained on human speech?

2. Can we systematically detect individual animal callers using said embedding spaces?

For this study we focus on marmosets (*Callithrix Jacchus*).
2. Study Design
Dataset

- We used a marmoset dataset collected and labeled by a previous paper\(^1\).
- Contains audio vocalization segments of with \texttt{call-type} and \texttt{caller identities} labels.
- 73k vocalization segments (7.7 hours).
- Task: caller detection.

![Vocalization per callers grouped by call-type.](image)

![Log distribution of vocalization lengths for callers 1-10.](image)

## Embedding Spaces

- 11 selected SSL models.

<table>
<thead>
<tr>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>APC</td>
</tr>
<tr>
<td>VQ-APC</td>
</tr>
<tr>
<td>NPC</td>
</tr>
<tr>
<td>Mockingjay</td>
</tr>
<tr>
<td>TERA</td>
</tr>
<tr>
<td>Mod-CPC</td>
</tr>
<tr>
<td>Wav2Vec2</td>
</tr>
<tr>
<td>Hubert</td>
</tr>
<tr>
<td>DistilHubert</td>
</tr>
<tr>
<td>WavLM</td>
</tr>
<tr>
<td>Data2Vec</td>
</tr>
</tbody>
</table>
Embedding Spaces

- 11 selected SSL models.
- Pre-trained on human speech.

<table>
<thead>
<tr>
<th>Model</th>
<th>Corpus</th>
</tr>
</thead>
<tbody>
<tr>
<td>APC</td>
<td>LS 360</td>
</tr>
<tr>
<td>VQ-APC</td>
<td>LS 360</td>
</tr>
<tr>
<td>NPC</td>
<td>LS 360</td>
</tr>
<tr>
<td>Mockingjay</td>
<td>LS 360</td>
</tr>
<tr>
<td>TERA</td>
<td>LS 360</td>
</tr>
<tr>
<td>Mod-CPC</td>
<td>LL 60k</td>
</tr>
<tr>
<td>Wav2Vec2</td>
<td>LS 360</td>
</tr>
<tr>
<td>Hubert</td>
<td>LS 360</td>
</tr>
<tr>
<td>DistilHubert</td>
<td>LS 360</td>
</tr>
<tr>
<td>WavLM</td>
<td>LS 360</td>
</tr>
<tr>
<td>Data2Vec</td>
<td>LS 360</td>
</tr>
</tbody>
</table>

LS refers to LibriSpeech, and LL is Libri-Light.
## Embedding Spaces

- 11 selected SSL models.
- Pre-trained on human speech.

<table>
<thead>
<tr>
<th>Model</th>
<th>Corpus</th>
<th>$P$</th>
<th>$D$</th>
</tr>
</thead>
<tbody>
<tr>
<td>APC</td>
<td>LS 360</td>
<td>4.11</td>
<td>512</td>
</tr>
<tr>
<td>VQ-APC</td>
<td>LS 360</td>
<td>4.63</td>
<td>512</td>
</tr>
<tr>
<td>NPC</td>
<td>LS 360</td>
<td>19.38</td>
<td>512</td>
</tr>
<tr>
<td>Mockingjay</td>
<td>LS 100</td>
<td>21.33</td>
<td>768</td>
</tr>
<tr>
<td>TERA</td>
<td>LS 100</td>
<td>21.33</td>
<td>768</td>
</tr>
<tr>
<td>Mod-CPC</td>
<td>LL 60k</td>
<td>1.84</td>
<td>256</td>
</tr>
<tr>
<td>Wav2Vec2</td>
<td>LS 960</td>
<td>95.04</td>
<td>768</td>
</tr>
<tr>
<td>Hubert</td>
<td>LS 960</td>
<td>94.68</td>
<td>768</td>
</tr>
<tr>
<td>DistilHubert</td>
<td>LS 960</td>
<td>27.03</td>
<td>768</td>
</tr>
<tr>
<td>WavLM</td>
<td>LS 960</td>
<td>94.38</td>
<td>768</td>
</tr>
<tr>
<td>Data2Vec</td>
<td>LS 960</td>
<td>93.16</td>
<td>768</td>
</tr>
</tbody>
</table>

LS is LibriSpeech, and LL is Libri-Light.  
$P$ indicates the number of parameters in millions.  
$D$ corresponds to the last layer embedding’s dimension.
## Embedding Spaces

- 11 selected SSL models.
- Pre-trained on human speech.
- 4 different pre-text tasks.

<table>
<thead>
<tr>
<th>Model</th>
<th>Corpus</th>
<th>(P)</th>
<th>(D)</th>
<th>Pretext Objective</th>
</tr>
</thead>
<tbody>
<tr>
<td>APC</td>
<td>LS 360</td>
<td>4.11</td>
<td>512</td>
<td>Autoreg. Recon.</td>
</tr>
<tr>
<td>VQ-APC</td>
<td>LS 360</td>
<td>4.63</td>
<td>512</td>
<td>Autoreg. Recon.</td>
</tr>
<tr>
<td>NPC</td>
<td>LS 360</td>
<td>19.38</td>
<td>512</td>
<td>Masked Recon.</td>
</tr>
<tr>
<td>Mockingjay</td>
<td>LS 100</td>
<td>21.33</td>
<td>768</td>
<td>Masked Recon.</td>
</tr>
<tr>
<td>TERA</td>
<td>LS 100</td>
<td>21.33</td>
<td>768</td>
<td>Masked Recon.</td>
</tr>
<tr>
<td>Mod-CPC</td>
<td>LL 60k</td>
<td>1.84</td>
<td>256</td>
<td>Contrastive</td>
</tr>
<tr>
<td>Wav2Vec2</td>
<td>LS 960</td>
<td>95.04</td>
<td>768</td>
<td>Contrastive</td>
</tr>
<tr>
<td>Hubert</td>
<td>LS 960</td>
<td>94.68</td>
<td>768</td>
<td>Masked Pred.</td>
</tr>
<tr>
<td>DistilHubert</td>
<td>LS 960</td>
<td>27.03</td>
<td>768</td>
<td>Masked Pred.</td>
</tr>
<tr>
<td>WavLM</td>
<td>LS 960</td>
<td>94.38</td>
<td>768</td>
<td>Masked Pred.</td>
</tr>
<tr>
<td>Data2Vec</td>
<td>LS 960</td>
<td>93.16</td>
<td>768</td>
<td>Masked Pred.</td>
</tr>
</tbody>
</table>

LS is LibriSpeech, and LL is Libri-Light.

\(P\) indicates the number of parameters in millions.

\(D\) corresponds to the last layer embedding’s dimension.
3. Caller Discrimination Analysis
Research Questions

We design a study with the following research questions:

1. How discriminative are the embedding spaces of SSL models pre-trained on human speech?

2. Can we systematically detect individual Marmoset callers using said embedding spaces?
Pipeline

Raw Audio Signal $s$

Marmoset vocalizations. Variable length segment.

SSL DNN $\mathcal{F}_k$

Pre-trained on human speech with different objective functions.

Extracted Embeddings $X$

Last FC layer. Variable-length.

Contrastive Masked Reconstruction

Masked Reconstruction

Autoregressive Reconstruction

Masked Prediction

$\mathbb{R}^{N \times D}$
Caller Groups

Train Embed $\chi$
Sort the embeddings by caller to:
- Effectively model each caller while accounting for the low vocalization segment length.
- Explore the acoustic variations within each caller.
Caller Groups

CID 1
- G1
- ... G100

CID 2
- G1
- ... G100

... CID 10
- G1
- ... G100

Train Embed

Arrow labeled "Sort" pointing to CID 1, CID 2, and CID 10.
We model the embedding spaces of each caller-group with a multivariate Gaussian distribution.
Pairwise Distances

We model the embedding spaces of each caller-group with a multivariate Gaussian distribution.
Pairwise Distances

Train Embed $\mathcal{X}$

Sort

CID 1

$\mathcal{N}(\mu, \Sigma)$

$N(\mu, \Sigma)$

$N(\mu, \Sigma)$

$N(\mu, \Sigma)$

G1

G100

Intra-group distances

CID 2

$\mathcal{N}(\mu, \Sigma)$

$N(\mu, \Sigma)$

$N(\mu, \Sigma)$

$N(\mu, \Sigma)$

G1

G100

Intra-group distances

$\binom{100}{2} = 4950$ distances

CID 10

$\mathcal{N}(\mu, \Sigma)$

$N(\mu, \Sigma)$

$N(\mu, \Sigma)$

$N(\mu, \Sigma)$

G1

G100

Intra-group distances
Pairwise Distances

Inter-group distances

Train Embed

\[ \mathcal{X} \]

Sort

CID 1

CID 2

CID 10

Split

G1

G100

G1

G100

G1

G100

100 \cdot 100 = 10000 

Inter-group distances

\[ \mathcal{N}(\mu, \Sigma) \]

100 \cdot 100 = 10000 

\[ \mathcal{N}(\mu, \Sigma) \]
Pairwise Distances

Distance measures between two multivariate Gaussian distributions $\mathcal{N}_f$ and $\mathcal{N}_g$:

- **KL-Divergence:**

$$D_{KL}(f||g) = \frac{1}{2} \left( \log \frac{||\Sigma_g||}{||\Sigma_f||} + \text{Tr}(\Sigma_f^{-1}\Sigma_g) + (\mu_f - \mu_g)^T \Sigma_g^{-1} (\mu_f - \mu_g) - d \right)$$

- **Bhattacharyya Distance:**

$$D_{BC}(f||g) = \frac{1}{8} (\mu_f - \mu_g)^T \Sigma_f^{-1} (\mu_f - \mu_g) + \frac{1}{2} \log \left( \frac{||\Sigma||}{\sqrt{||\Sigma_f|| \cdot ||\Sigma_g||}} \right)$$

---

Results

- We can visualize the computed distribution of distances through a distance matrix:
  - Diagonal: intra-caller group distances
  - Off-diagonal: inter-caller group distances.
  - Darker regions indicate higher dissimilarity.

- Ideal scenario: the intra-class distances to be smaller than the inter-class ones.

- Not entirely the case in our results.
Results

- Nevertheless, for callers with a larger amount of available data (Caller 1-3), we observe good discrimination when compared to callers with a lower amount of data (Caller 8).

- Analysis suggests that the SSL embeddings do carry information for distinguishing marmoset callers to a certain extent.

- Accomplishing this with a simple linear classifier may still be a challenging task.
4. Caller Detection Study
Research Questions

We design a study with the following research questions:

1. How discriminative are the embedding spaces of SSL models pre-trained on human speech?

2. Can we systematically detect individual Marmoset callers using said embedding spaces?
Caller Detection Pipeline

- **Raw Audio Signal \( s \)**
  - Marmoset vocalizations.
  - Variable length segment.

- **SSL DNN \( \mathcal{F}_k \)**
  - Pre-trained on *human speech* with different objective functions.

- **Extracted Embeddings \( x := \mathcal{F}_k(s) \)**
  - Last layer.
  - Variable-length.
  - \( \mathbb{R}^{N \times D} \)

- **Contrastive**
- **Autoregressive Reconstruction**
- **Masked Reconstruction**
- **Masked Prediction**
Caller Detection Pipeline

- **Raw Audio Signal $s$**
- **SSL DNN $\mathcal{F}_k$**
  - Pre-trained on *human speech* with different objective functions.
  - Last layer. Variable-length.
- **Extracted Embeddings $x := \mathcal{F}_k(s)$**
- **Functionals $f$**
  - Concatenated statistics of the embedding across $N$. Fixed-length.

Marmoset vocalizations. Variable length segment.

- **Contrastive**
- **Autoregressive Reconstruction**
- **Masked Reconstruction**
- **Masked Prediction**

Functionals $f$

- $\mathbb{R}^{N \times D}$
- $(\text{Mean } \mu, \text{Diagonal } \Sigma)$
- $\mathbb{R}^{2D}$
**Caller Detection Pipeline**

- **Raw Audio Signal** $s$
  - Marmoset vocalizations. Variable length segment.

- **SSL DNN** $\mathcal{F}_k$
  - Pre-trained on *human speech* with different objective functions.

- **Extracted Embeddings** $x := \mathcal{F}_k(s)$
  - Last layer. Variable-length.

- **Functionals** $f$
  - Concatenated statistics of the embedding across $N$. Fixed-length.

- **Classifier**
  - SVMs, RF, AB

- **Contrastive**
- **Autoregressive Reconstruction**
- **Masked Reconstruction**
- **Masked Prediction**

- Output: $\mathbb{R}^{N \times D}$ (Mean $\mu$, Diagonal $\Sigma$) $\mathbb{R}^{2D}$
**Caller Detection Pipeline**

- **Raw Audio Signal** \( s \)
- **SSL DNN** \( \mathcal{F}_k \)
- **Extracted Embeddings** \( x := \mathcal{F}_k(s) \)
- **Functionals** \( f \)
- **Classifier**

- Pre-trained on *human speech* with different objective functions.
- Last layer. Variable-length.
- Concatenated statistics of the embedding across \( N \). Fixed-length.

- SVMs, RF, AB

- Marmoset vocalizations. Variable length segment.

- Contrastive
- Autoregressive Reconstruction
- Masked Reconstruction
- Masked Prediction

- \( \mathbb{R}^{N \times D} \)

- (Mean \( \mu \), Diagonal \( \Sigma \))

- \( \mathbb{R}^{2D} \)

- Accept or reject

- The claimed identity
Caller Detection Study
## Caller Detection Study

Classifiers:
- Random Forest (RF).
- Support Vector Machine (SVM).
- AdaBoost (AB).

<table>
<thead>
<tr>
<th>Classifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>RF</td>
</tr>
<tr>
<td>AB</td>
</tr>
<tr>
<td>SVM</td>
</tr>
<tr>
<td>L SVM</td>
</tr>
</tbody>
</table>
## Caller Detection Study

### Classifiers:
- Random Forest (RF).
- Support Vector Machine (SVM).
- AdaBoost (AB).

### Framework:
- 5-fold cross-validation.
- Hyper-parameter tuning on each fold.
- Grid Search.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Hyperparameters</th>
<th>Search space</th>
</tr>
</thead>
<tbody>
<tr>
<td>RF</td>
<td># Estimators</td>
<td>[50, 500, 1000, 2000]</td>
</tr>
<tr>
<td></td>
<td>Max # Features</td>
<td>{'auto', 'sqrt', 'log2'}</td>
</tr>
<tr>
<td></td>
<td>Criterion</td>
<td>['gini', 'entropy']</td>
</tr>
<tr>
<td></td>
<td>Min samples leaf</td>
<td>[1, 2, 4]</td>
</tr>
<tr>
<td>AB</td>
<td>Learning rate</td>
<td>[0.1, 0.2, 0.5, 1]</td>
</tr>
<tr>
<td></td>
<td>Algorithms</td>
<td>[SAMME, SAMME.R]</td>
</tr>
<tr>
<td></td>
<td>Max # Estimators</td>
<td>[50, 500, 1000, 2000]</td>
</tr>
<tr>
<td>SVM</td>
<td>C</td>
<td>1e[-5, -4, -3, -2, -1, 0]</td>
</tr>
<tr>
<td></td>
<td>Kernel</td>
<td>[RBF, Linear, Polynomial]</td>
</tr>
<tr>
<td></td>
<td>Gamma</td>
<td>['scale', 'auto']</td>
</tr>
<tr>
<td>LSVM</td>
<td>C</td>
<td>1e[-5, -4, -3, -2, -1, 0]</td>
</tr>
<tr>
<td></td>
<td>Max # Iterations</td>
<td>10000</td>
</tr>
<tr>
<td></td>
<td>Class weights</td>
<td>['balanced', 'None']</td>
</tr>
</tbody>
</table>

Search space to find optimal hyper-parameters.
Caller Detection Study

Classifiers:
- Random Forest (RF).
- Support Vector Machine (SVM).
- AdaBoost (AB).

Framework:
- 5-fold cross-validation.
- Hyper-parameter tuning on each fold.
- Grid Search.

Task: Caller detection.
Performance measure: AUC.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Hyperparameters</th>
<th>Search space</th>
</tr>
</thead>
<tbody>
<tr>
<td>RF</td>
<td># Estimators</td>
<td>[50, 500, 1000, 2000]</td>
</tr>
<tr>
<td></td>
<td>Max # Features</td>
<td>{'auto', 'sqrt', 'log2'}</td>
</tr>
<tr>
<td></td>
<td>Criterion</td>
<td>['gini', 'entropy']</td>
</tr>
<tr>
<td></td>
<td>Min samples leaf</td>
<td>[1, 2, 4]</td>
</tr>
<tr>
<td>AB</td>
<td>Learning rate</td>
<td>[0.1, 0.2, 0.5, 1]</td>
</tr>
<tr>
<td></td>
<td>Algorithms</td>
<td>[SAMME, SAMME.R]</td>
</tr>
<tr>
<td></td>
<td>Max # Estimators</td>
<td>[50, 500, 1000, 2000]</td>
</tr>
<tr>
<td>SVM</td>
<td>C</td>
<td>1e[-5, -4, -3, -2, -1, 0]</td>
</tr>
<tr>
<td></td>
<td>Kernel</td>
<td>[RBF, Linear, Polynomial]</td>
</tr>
<tr>
<td></td>
<td>Gamma</td>
<td>['scale', 'auto']</td>
</tr>
<tr>
<td>LSVM</td>
<td>C</td>
<td>1e[-5, -4, -3, -2, -1, 0]</td>
</tr>
<tr>
<td></td>
<td>Max # Iterations</td>
<td>10000</td>
</tr>
<tr>
<td></td>
<td>Class weights</td>
<td>['balanced', 'None']</td>
</tr>
</tbody>
</table>

Search space to find optimal hyper-parameters.
Results and Discussion

- SVM classifier gives the best performance across all embedding spaces.
- The decision tree-based ensemble methods exhibit comparable performance for most models, consistently outperform LSVM.

<table>
<thead>
<tr>
<th>Model</th>
<th>AB</th>
<th>LSVM</th>
<th>RF</th>
<th>SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>APC</td>
<td>71.44</td>
<td>65.18</td>
<td>70.89</td>
<td>79.16</td>
</tr>
<tr>
<td>VQ-APC</td>
<td>71.60</td>
<td>65.58</td>
<td>70.04</td>
<td>78.45</td>
</tr>
<tr>
<td>NPC</td>
<td>72.61</td>
<td>66.27</td>
<td>71.50</td>
<td>77.32</td>
</tr>
<tr>
<td>Mockingjay</td>
<td>72.39</td>
<td>64.43</td>
<td>71.75</td>
<td>78.44</td>
</tr>
<tr>
<td>TERA</td>
<td>70.34</td>
<td>64.57</td>
<td>68.43</td>
<td>74.03</td>
</tr>
<tr>
<td>Mod-CPC</td>
<td>72.62</td>
<td>64.05</td>
<td>69.81</td>
<td>75.96</td>
</tr>
<tr>
<td>Wav2Vec2</td>
<td>74.41</td>
<td>63.94</td>
<td>70.18</td>
<td>75.85</td>
</tr>
<tr>
<td>Hubert</td>
<td>71.71</td>
<td>64.14</td>
<td>70.17</td>
<td>75.64</td>
</tr>
<tr>
<td>DistilHubert</td>
<td>70.77</td>
<td>65.11</td>
<td>70.34</td>
<td>76.26</td>
</tr>
<tr>
<td>WavLM</td>
<td>73.97</td>
<td>65.32</td>
<td>70.74</td>
<td>78.60</td>
</tr>
<tr>
<td>Data2Vec</td>
<td>69.81</td>
<td>62.58</td>
<td>68.23</td>
<td>73.04</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td><strong>71.97</strong></td>
<td><strong>64.66</strong></td>
<td><strong>70.19</strong></td>
<td><strong>76.61</strong></td>
</tr>
</tbody>
</table>

Macro AUC scores [%] on Test with 5-fold CV.
Results and Discussion

- Per-caller detection performance in distinguishing a positive class from the negative instances using SVM on a single Test fold.

  ➡ All callers are systematically distinguished in this binary framework, including the classes with a low amount of data (CID 6–8).

AUC-ROC curves per caller class (CID) for WavLM embeddings using RBF SVM on one fold of Test.
Results and Discussion

- SVM’s average performance for each embedding space across the 5 folds.
- Embedding spaces of all models are capable of successfully differentiating Marmoset callers.
- Indicates that SSL models pre-trained on human speech data can generate salient representations capable of distinguishing animal vocalizations regardless of the pre-training criterion.

Macro average ROC curves of all models on Test using RBF SVM over all folds. Shaded areas represent ± 1 std over the 5-folds.
Results and Discussion

- Relationship between the number of parameters and detection performance for all models.
- No clear pattern.
  - WavLM’s embedding space is more separable than the other masked prediction models.
  - Both auto-regressive reconstruction models perform exceptionally well with significantly fewer parameters.
  - All pre-training criteria yields competitive performance, some are more efficient than others, allowing smaller models to perform comparably to larger models.
Summary

- **Aim:** we investigated the applicability of SSL representations, pre-trained on human speech, to analyze animal vocalizations.

- **Findings:** such representations are capable of classifying vocalizations in the bio-acoustics domain for tasks such as Marmoset caller detection.

- **Consequence:** findings can greatly benefit bio-acoustics researchers looking to distinguish individual identities within a specific species in their acoustic data.
Open-Ended Questions

● We only looked at DNNs pre-trained on human speech. What about ones trained on generic audio sets?
● We only looked at embedding spaces. What about traditional spectral features?
● We only evaluated linear probing. What about fine-tuning on a downstream task?
● We only looked at marmosets. What about other animals?
Thank you!

Idiap Research Institute

https://github.com/idiap/sslCallerDetection

eklavya.sarkar@idiap.ch
FAQ - Lack of Baseline

- There are ‘no’ prior works in the literature that guide us as to which features are important for caller detection: no baseline feature and approach exist for the task at hand.
- We took inspiration from speech processing, where embedding-based speaker verification systems are becoming the norm, and investigated with SSL neural embedding representations.
- Some SSL models are trained with log-mel spectral features as input.
- Our implemented methods can now serve as baselines. 😊
- Our focus was not to achieve the best performance.
FAQ - Lack of Baseline

- **MFCC:**
  - Window size: 15 ms (240 samples)
  - Window shift: 5 ms (80 samples)

- Weaker performance compared to pre-trained SSL models.
FAQ - Bias towards certain call-types

- We are constrained by the scarcity of certain call-type classes in the dataset.
- Due to limited data availability, we can’t comprehensively investigate this question.
- Even if such a study were carried out on this dataset, it would be challenging to conclude whether the observed differences in performance were due to call-types or data scarcity.
- This issue does not change or invalidate the analysis and findings in our paper.