

研究進捗報告

竹本志恩

July 4, 2025

研究背景

これまでの研究背景

- 少子高齢化による介護負担の増大
- 安価で手軽な見守りシステムの需要
 - 迅速な対応
 - 監視の負担軽減

研究背景に対する疑問

- 施設での運用で介護者の負担は減る？
 - 本当にそれがやりたいことなのか？
- 「安価＋手軽さ」だけで、既存の技術と差別化できる？
 - 他のセンサー（カメラ、ウェアラブル等）と比較して本当に手軽？
 - なぜ「音」を選ぶのか、理由が必要
- なぜ機械学習を用いる？
 - HAR（Human Activity Recognition）には、より古典的で、適用範囲は狭いが確実性の高い手法も存在するのでは？

以下を念頭に置き，より深掘り

- AAL (Ambient Assisted Living) を汲む
 - 自立した生活を支援
 - 在宅での介護/医療を視野に
 - 施設より在宅の方が自立に近づく
 - 問題の早期発見で医療/介護負担を軽減
 - 費用や労働量など，高齢者とサービス提供者の負担を軽減
- 音特有の異常行動兆候の探求
 - 音独自の健康指標に着目
 - 呼吸音や咳き込みなど

- 機械学習を用いる理由
 - 環境への適応
 - 従来法は閾値で判断; 適応的な手法も存在
 - 機械学習は柔軟で多様な環境に適応可能
 - 在宅見守りへの応用
 - 支援があれば自立生活が可能な高齢者を対象に考える
 - 単一目的の検知より, 様々な異常の兆候を柔軟に検知したい

研究計画

仮ロードマップ

- 今後2週間で計画を具体化
- 7月: 基礎の準備
 - 背景サーベイ
 - マルチラベル音響イベント検知モデルの実装
 - データ収集
- 8月: 必要な施策の実行
 - 半教師あり連合学習の調査・実装
 - シミュレーション方法の調査・検討
- 9月: 異常検知層の追加
- 10月: FLの実装・実験

具体化のための計画

- 以下を検証し, 研究計画の目処を付ける
- 主要タスク
 - ✓ 基礎モデル構築
 - データ収集
 - データ不均衡対策
 - 半教師あり連合学習
 - 異常検知モデルの構築
 - シミュレーションの検討

現在の進捗

実装: マルチラベル音響検知モデル (PoC)

- モデル: CNN → Transformer ベース
- データ生成:
 - ESC-50 データセットの一部カテゴリを使用
 - 2つの音の組み合わせを全パターン作成 (約 12,000)
 - ラベルの組が均衡になるよう 9,048 データを選定
- 使用ラベルは雨やドアのノック音など (cf. 補足)
- 日常生活の基本的な音を想定

学習

- 選択した 9,048 データで訓練
- パラメータなど: 補足へ
- マルチラベル用の構成
 - * 出力: Sigmoid 関数
 - * 最適化アルゴリズム: Adam
 - * 損失関数: バイナリクロスエントロピー

評価

- 訓練時と異なる 936 データを使用
- 各精度指標

Micro Precision/Recall/F1
= 0.9655 / 0.8682 / 0.9143
Macro **F1** = 0.8458

サーベイ：研究背景の再考

- 現状の反省：技術ありきで話を進めており、研究意義の説明が不足
- 対応：
 - 関連論文を読み、研究背景を深掘り中
 - サーベイ論文を 3 本程度確認済み
 - 今後、サーベイ論文から引用されている重要論文を精読予定

データ収集: 異常イベントの音響データ

- 音特有の異常（呼吸、咳き込み、悲鳴など）に関するデータを収集したい
- 調査済みデータセット
 - Deeply Nonverbal Vocalization Dataset: 作者への連絡が必要
 - Respiratory Sound Database:
 - デジタル聴診器の音源が多く、行動認識には不向きか
 - 'AKG C417L Microphone'の音源は使える可能性あり
 - ライセンスが不明
- 未確認データセット候補
 - SAFE: 転倒音
 - TAME Pain Dataset: 痛みに関連する音声
 - Sound-Dr Dataset: おそらく呼吸音
 - F2LCough: 咳の分類

展望と課題

- 各種対応の簡易的な試行 (所要時間とタスク内容の把握)
 - データ不均衡対応: Focal Loss などの導入を検討
 - 半教師あり連合学習の調査, 検証
 - 異常検知モデルの構築
 - シミュレーション環境の検討
- AAL サーベイの継続
 - 類似研究を調査し、研究の新規性・貢献を明確に
- データ収集と分析
 - 各種データセットの内容を精査
 - 音で検知可能な異常の兆候を洗い出す

- 懸念事項: 自作データセットの評価方法
 - 構築したデータセットでモデルをどう評価する？
 - どうすれば実用に耐えうるモデルだと示せるか
 - マルチラベルデータ作成時, 現実的なラベル設計が重要そう

補足

- 使用ラベル (ESC-50 より抜粋)

```
allowed_categories = [  
    "rain", "door_wood_knock", "door_wood_creaks",  
    "glass_breaking", "sneezing", "breathing", "coughing",  
    "footsteps", "laughing", "brushing_teeth", "snoring",  
    "drinking_sipping", "pouring_water", "toilet_flush"  
]
```

学習の Hyperparameters:

```
model = MelSpectrogramTransformer(  
    input_dim=n_mels,  
    embed_dim=128,          # 埋込次元  
    nhead=4,                # Attn ヘッド数  
    nhid=256,               # FFN 隠れ層  
    nlayers=2,              # Encoder 層数  
    n_classes=num_classes,  
    max_len=X.shape[2])
```

実行コマンド例:

```
uv run model_module/train.py \  
    --data ./dataset/esc50_multilabel.npz \  
    --batch 32 \  
    --epochs 20 \  
    --lr 1e-3 \  
    --val-split 0.1 \  
    --device cuda \  
    --save-model ./checkpoints/label-14_transformer.pth
```

評価実行コマンド例:

```
uv run ./model_module/eval.py \  
    --data ./dataset/esc50_multilabel.npz \  
    --model ./checkpoints/label-14_transformer.pth \  
    --batch 64 \  
    --device mps \  
    --threshold 0.5
```

[RESULT]

```
Threshold = 0.50  
Micro Precision/Recall/F1  
= 0.9655 / 0.8682 / 0.9143  
Macro F1 = 0.8458
```

補足: Classification Report (per class):

	precision	recall	f1-score	support
Class rain	0.94	0.84	0.89	19
Class door_wood_knock	1.00	0.85	0.92	20
Class door_wood_creaks	1.00	0.83	0.91	18
Class glass_breaking	1.00	0.80	0.89	20
Class sneezing	0.89	0.85	0.87	20
Class breathing	1.00	0.55	0.71	20
Class coughing	0.89	0.96	0.92	25
Class footsteps	1.00	0.93	0.97	15
Class laughing	1.00	0.88	0.93	16
Class brushing teeth	1.00	1.00	1.00	25
Class snoring	0.95	0.95	0.95	19
Class drinking_sipping	0.96	0.88	0.92	25
Class pouring_water	1.00	0.94	0.97	16
Class toilet_flush	0.00	0.00	0.00	0
micro avg	0.97	0.87	0.91	258
macro avg	0.90	0.80	0.85	258
weighted avg	0.97	0.87	0.91	258
samples avg	0.98	0.87	0.90	258

補足: 研究背景

- サーベイ前に書いた研究背景
- 今後サーベイを通じて詰める

