

オフィス業務把握のための音声データ取得・行動分析システム SLeSAS のプロトタイプの実装と予備評価

檜田 雄斗^{†1} 栗飯島 陽向^{†1} 大西 裕也^{†2} 大住 卓史^{†3} 武本 充治^{†1}

概要：多様な人間が活動するオフィスでは、その業務を向上させるために補助や助言を与えることが必要であるが、適切なタイミングの決定が難しいという問題がある。この解決策として、主たる業務を邪魔しない方法として、小型コンピュータによる音声データの取得と行動分析するシステム SLeSAS(Supervised Learning Source Analysis System)のプロトタイプの実装を行った。SLeSAS プロトタイプ実装を実際のオフィスで動作させることにより、性能に関する課題点が明らかになった。

1. はじめに

企業・官庁などのオフィスには様々な人がおり、その経験や知識、理解能力は多様である。それらの人が集団として効率的に業務をこなすためには、各担当者に対して補助や助言を与えることが重要であるのは当然であるが、そのタイミングをどのように決定するのか難しい問題である。これは補助や助言が欲しいと担当者が明示的に伝える場合がほとんどであることに起因している。

この問題を解決するためのシステム導入で最も重要なのは、主たる業務に影響が少ない方法で、担当者の状況が把握できることである。つまり、状況把握のために担当者が追加の操作を行わない方式を採用することが重要である。

そこで我々はオフィスで常に生成されているデータである音声に注目した。オフィスにいる人の声の他、オフィスにある OA 機器からは常に音声が出ている。特に OA 機器と担当者が触れているキーボードやマウスからは様々な音が発生している。

また、音声から行動を分析するにあたって大規模なコンピュータシステムをオフィスに設置することは難しいため、小型コンピュータをオフィスに置くことで音声データ取得と行動分析をするシステム SLeSAS を実装することとした。

SLeSAS を用いれば、担当者の行動分析が上司や同僚に把握することが可能になり、「この人は今困っているのかな...、だから声をかけよう」という人の行動を変容させることも期待できる。さらに、このようなオフィスのメンバの把握により、全体のマネジメント効率が良くなることも期待できる。

本稿では、オフィスでの業務を把握するための音声データからの行動分析システム SLeSAS の基本機能、拡張機能、プロトタイプ実装と動作により明らかになった問題点について述べる。

2. SLeSAS の概要

オフィスという実空間の情報をセンサで取得し、それを分析することで、実際に業務に従事しているオフィスワーカーの行動を分析することで、業務推進・改善に必要な補助や助言を与えるシステムを実現するために重要なことは通常の業務に影響を与えない方法を採用することである。

この点に注目し、我々はセンサとしては音声センサを使用し、データを処理するコンピュータとしては小型の Raspberry Pi を用いることで、どこまでデータ取得と行動分析ができるかを検証する。分析結果はマネージャが把握することができ、マネージャは分析結果から「社員の作業が難航しているのではないか」といった判断が可能になるため、必要に応じてマネージャは補助や助言を与えることができる (図 1)。

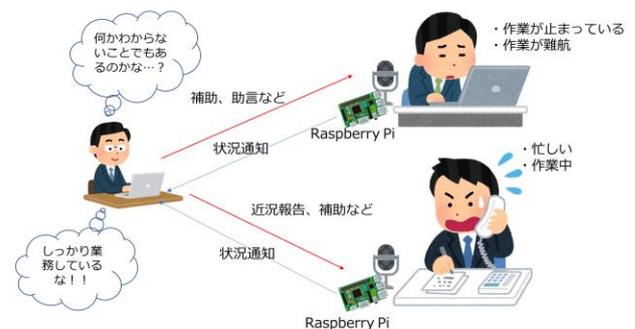


図 1: オフィスにおけるデータ取得と行動分析

SLeSAS を実現するプログラムは Raspberry Pi 上での Python で実装される。音声データの取得 (録音) 部分と行動分析 (推論) 部分に分けられ、本システムではこれらの動作を並列に処理を行っていく。また、音声データは Raspberry Pi の中の wav 形式、推論する際の画像データに関しては png 形式のファイルとして、取り扱われる (図 2)。

^{†1} 東京国際工科専門職大学

^{†2} ICSP idea 株式会社

^{†3} 株式会社 ICS パートナーズ

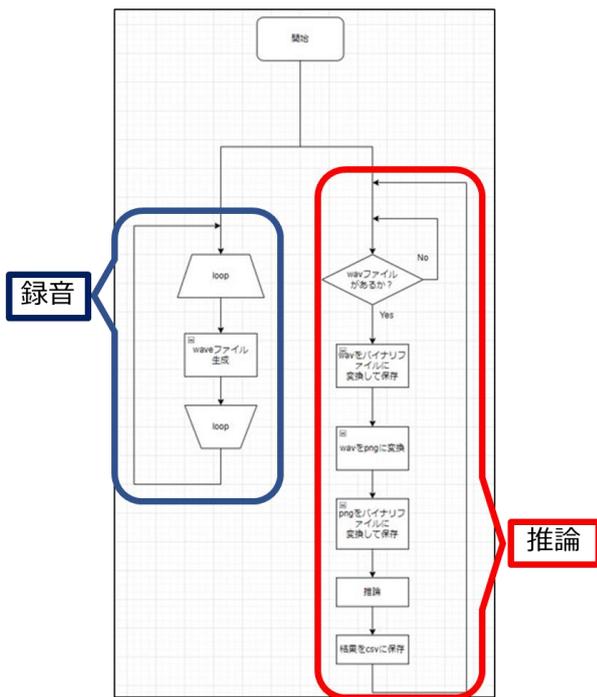


図 2: 基本機能フロチャート

3. 基本機能

現在のシステム実装と動作はあらかじめ業務を行う際発生する環境音,例えば「キーボードを打つ音」,「マウスを操作する音」,「無音」などと言った音声データの周波数画像を機械学習させ,生成したモデルを用いてリアルタイムで 10 秒間隔の音声データを画像にして判断させる。

図 3, 4, 5 に SLeSAS プロトタイプの録音機能を使って取得した音声データの周波数を画像データに変換して表示したものを示す。

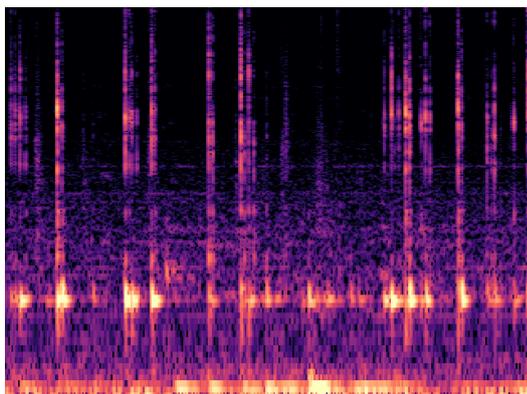


図 3: キーボード操作時の周波数画像

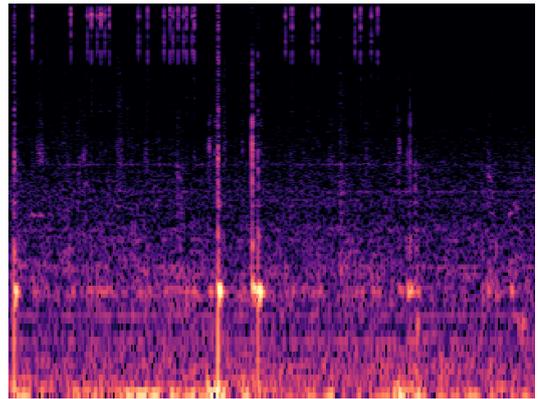


図 4: マウス操作時の周波数画像

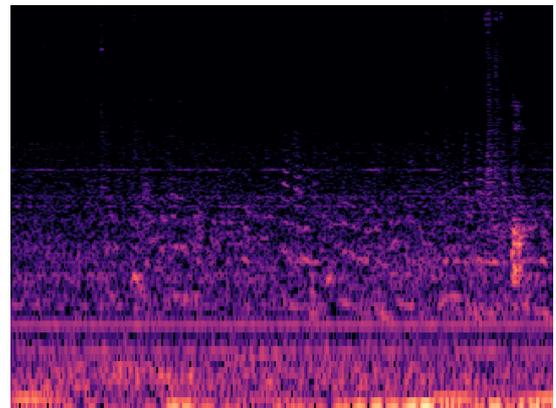


図 5: 無音状態の周波数画像

図 3 はキーボードを操作している時の画像で,小さな波形が連続して検出されており,実際の動作では連続で打鍵していることが確認できる。

図 4 はマウスを操作している画像で小さな波形が連続で二回検出されており,実際の動作ではマウスでダブルクリックをしていることが確認できる。

図 5 は波形が一切確認できないため,無音状態,つまりは何もしていないことが確認できる。

4. SLeSAS の動作状況

SLeSAS プロトタイプの基本機能を実行したときにターミナルに出力されたデバッグ情報からプログラムの動作状況を把握することができる(図 6)。

```

デバック Predict_Record filename : ./20221110-101338.wav
デバック WavConvert: ./20221110-101338.wav
デバック Predict_Record png_filename : ./20221110-101338.png
デバック PngConvert: ./20221110-101338.png
Thinking time was: 0
デバック LogOutPut : 20221110-101338
デバック LogOutPut CSV_File : Predict_Log.csv
file name : ./20221110-101338.png
分類クラス : ['key_touch', 'mouse', 'noise']
確率 : [[99.991 0.009 0. ]]
判定結果 : key_touch
作業内容 : Using Keyboard!

-----

デバック Predict_Record filename : ./20221110-101349.wav
デバック WavConvert: ./20221110-101349.wav

デバック WAVE_OUTPUT_FILENAME : 20221110-101400
デバック Predict_Record png_filename : ./20221110-101349.png
デバック PngConvert: ./20221110-101349.png
デバック LogOutPut : 20221110-101349
デバック LogOutPut CSV_File : Predict_Log.csv
file name : ./20221110-101349.png
分類クラス : ['key_touch', 'mouse', 'noise']
確率 : [[ 0. 0. 100.]]
判定結果 : noise
作業内容 : Thinking

```

図 6: VisualStudioCode での実行結果

図 6 は実際にプログラムを実行した際のターミナルの画像で、判別結果からキーボード操作と無操作状態であることが分かる。

また、判別結果、判別結果をもとにした利用者の動作内容、そして無音状態とその合計時間を CSV ファイルとして保存する機構も実装した(図 7)。

key_touch	Using Keyboard!	
noise	Thinking	Thinking
noise	Thinking	↓
mouse	Using Mouse!	60sec
noise	Thinking	Thinking
noise	Thinking	↓
noise	Stopping...	↓

図 7: CSV ファイルに保存されたログ

図 7 では、10 秒間隔で録音した音声データを画像に変換してその画像を” key_touch” ,” mouse” ,” noise” の 3 つに分類させた結果を表示させている。

5. 拡張機能の実装

オフィスでの活用を想定して、基本機能に追加すべき機能を選定し、その選定が正しいことを検証するために、実装を行った。

- 追加機能 1. 長時間離席した際、その人の離席状況を LINE などに知らせる機能
- 追加機能 2. プログラム実行時に出勤、退勤を CSV ファイルに自動的に書き込む機能
- 追加機能 3. Google カレンダーと連携して会議などの予定があった場合、音の記録ではなく、その予定をログに書き込む機能

これらの機能を基本機能の拡張で実装した(図 8)。

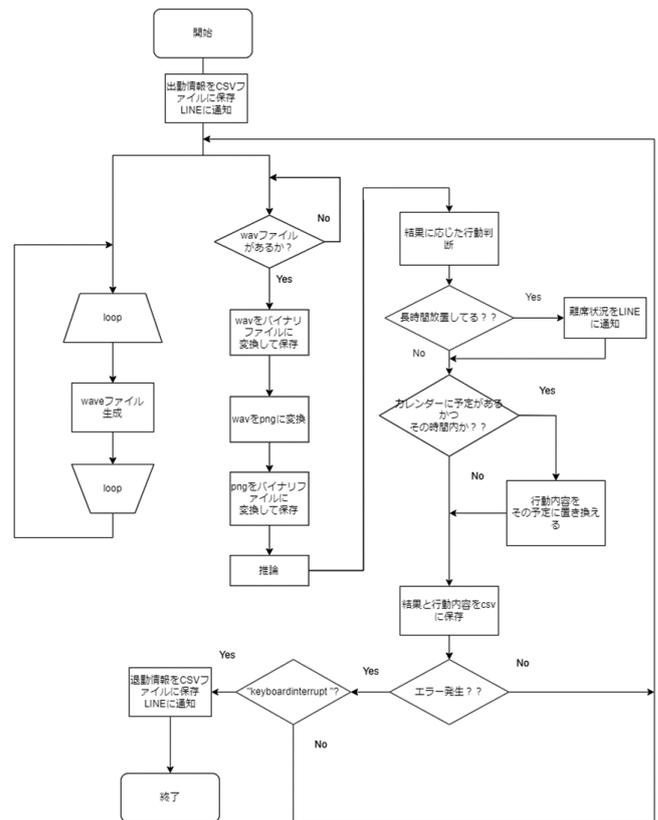


図 8: 拡張機能フローチャート

拡張機能 1 は、「リモートでも出勤時でも誰が放置、離席しているのかわかることによって仕事やプロジェクトの効率化が図れる」という我々の仮説に基づいている。

図 9 に拡張機能 1 の動作状況を示す。判別結果が”noise”でさらにそれが 10 分以上続く場合、10 分間隔で LINE に「〇〇離席中です」というメッセージを LINE Notify API[1] を用いて自動的に送信している。(今回は実験的に 10 分間隔で送信している)



図 9:離席状況を表示した LINE の画面

拡張機能 2 は出勤、退勤を別の CSV ファイルで保存するものである。システムを起動した際に自動的に出勤時間が記録され、システムを停止させると自動的に退勤時間と勤務時間が記録される。これは新規のソリューションではないが、既存のソリューションシステムの改良に寄与するという我々の仮説に基づいている。

図 10 に実際に生成する CSV ファイルの中身を示す。

11/16-11:22:06	業務開始:11/16-11:22:06	
11/16-11:22:09	業務終了:11/16-11:22:09	0:00:00.822665
11/16-11:22:41	業務開始:11/16-11:22:41	
11/16-11:22:43	業務終了:11/16-11:22:43	0:01:52.106325

図 10:出勤、退勤時間を書き込んだ CSV ファイル

拡張機能 3 は Google Calendar API[2]を用いてカレンダーに予定を入れた場合(今回は”会議”),音の判別結果がどのようなものでもその時間内は予定を書き込むといった機能である。この機能を追加することで、「本来事情があって離席していたはずなのにログでは“離席”と書き込まれてしまうことを防止」でき、適切な行動分析に寄与するという我々の仮説に基づいている。現在検証を行ったシステムでは PC 上で予定を入れたが、スマートフォン上の Google Calendar でも予定を入れられるのでかなり汎用性は高いと言える。

図 11 に予定が入った CSV ファイルを示す。

E	F	G
noise	Thinking	↓
noise	Stopping...	↓
mouse	Using Mouse!	70sec
noise	Thinking	Thinking
noise	Thinking	↓
noise	Thinking	↓
noise	Thinking	会議
noise	Stopping...	会議

図 11: 予定が入っている場合の CSV ファイル

6. 現実装における課題

基本機能・拡張機能を実装し、実際のオフィスに設置して動作を行ったところ、基本機能においていくつか課題点が見つかった。それらの内、性能に関して列挙すると以下のようになる。

1. 長時間の動作が不安定
2. 汎用的ではない
3. 判別速度
4. 確実に判別できるわけではない

まず、1. “長時間の動作が不安定”であるがこれは文字通り長時間、動作させた場合途中で止まってしまうということである。120 分連続で動作する時もあるが、短い時間で動作が止まってしまう場合がある。これは単に使用したデバイスのスペック不足やマルチタスクの負荷が原因の一つであると考えられる。実際に導入する際は予測する機械と判別する機械を分けると安定した動作が可能になるのではないかと考える。

次に 2. “汎用的ではない”であるが、これは利用する個人の環境によって結果が変動する可能性があるということである。例えば使用しているキーボードの音や環境音そういった要因が個人個人で異なるため判別結果にブレが生じる恐れがある。これの対応策として機械学習モデルを複数用意してそれぞれの結果をもとにして最終的な判別結果を出すのがよいと考える。

次に 3. “判別速度”についてだが、動作するに当たり録音が完了してから予測完了までに約 20 秒のラグがあるということである。つまり、結果が表示される場所には現在時刻の 20 秒くらい前の結果が表示されるということになっ

てしまうことになる。この解決案としては 1. “長時間の動作が不安定” で述べた解決案と同じく録音と判別を分けることが挙げられる。

最後に 4.” 確実に判別できるわけではない” について、個人の差にもよるが現段階での判別率は約 8 割であるため必然的に誤認識が発生する時がある。このことによって“放置していたはずなのにマウスの音が誤認識された” ということが起こりうる。この解決案としては特定の条件下の場合誤認識された結果を無効化するプログラムを作成する、もしくは誤認識されにくい精度の高いモデルを作成することで比較的安定した動作が可能になると考えられる。

7. まとめと今後の課題

本稿ではオフィス業務を把握するための音声データからの行動分析システム SLeSAS の基本機能と拡張機能のプロトタイプについて述べた。そのプロトタイプ実装を用いて、オフィスで予備実験を行ったところ、基本機能において、動作不安定、低い汎用性、遅い反応速度、低い判別可能性などの性能に関して改善すべき課題が明らかになった。また、LINE による通知、勤怠表に使えるデータの自動生成、Google Calendar との統合を拡張機能として実装を行った。これらの基本機能と拡張機能を実際のオフィスで動作させ、その結果を実際のマネージャに評価してもらうことで、本システムの改良を行うことが今後の課題である。

謝辞 本研究の実施にあたり、実際の測定環境を提供してくれた株式会社 ICS パートナース社員諸君に感謝する。

参考文献

- [1]LINE Notify. <https://notify-bot.line.me/doc/ja/>
(2022-12-19)
- [2]Google Calendar. [Google Calendar API | Google Developers](#)
(2022-12-19)