

FaceAPI を用いた表情分析を活用した 労務管理支援の提案

1932028 片平 純太郎

1932080 田口 真人

指導教員：中村 直人 教授

令和 4 年度

目次

第 1 章	はじめに	1
1.1	研究の背景と目的	1
1.2	論文の構成	2
第 2 章	機能設計	3
2.1	図の表示	3
第 3 章	システムの実装方法	4
3.1	システム構成	4
3.2	React について	5
3.3	face-api.js について	6
3.4	Material-UI について	6
3.5	Firebase Authentication	7
3.6	Cloud Firestore	8
第 4 章	Tex の書き方 (応用編)	9
4.1	応用コマンド・環境	9
4.2	マクロ	9
	謝辞	12
	参考文献	13
	参考文献	13

図目次

1.1	笑顔計測後の主な感情変化	1
1.2	厚生労働省によるパワハラ調査	2
3.1	システム構成図	4
4.1	図の名称	11

表目次

第1章

はじめに

1.1 研究の背景と目的

昨今、コロナの影響により、人と直接会う機会が減り、マスクの着用を強いられるようになった。その結果、笑顔になる機会が減るため、笑顔の減少に繋がる。

しかし、総合人材サービスのパーソルホールディングス株式会社が行った調査 [1] では、仕事をする上で笑顔になると「楽しい」という気持ちが高まった人は約6割で、ポジティブな感情状態で仕事に取り組んでいた人ほど、笑顔になっており、職場において笑顔が高まれば、自発的に取り組む傾向があるという結果が出た。

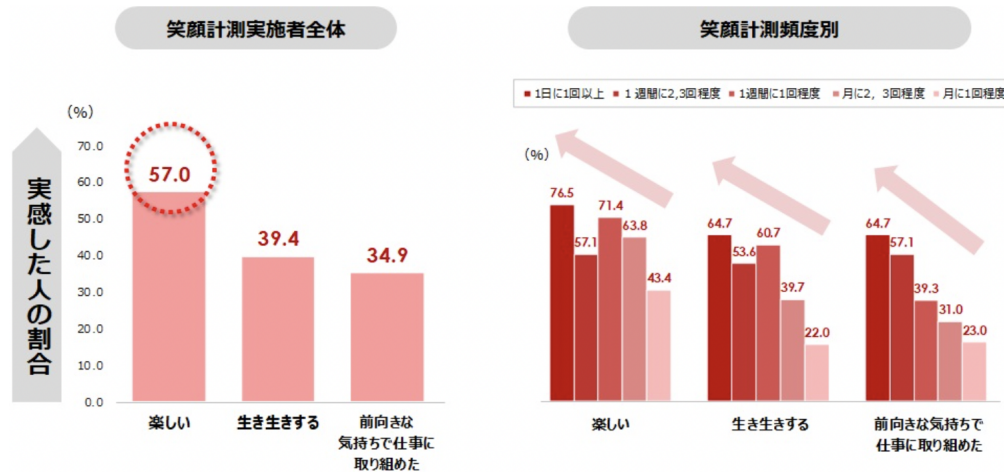


図 1.1 笑顔計測後の主な感情変化

また、厚生労働省の調査 [2] によると、下のグラフから分かるように、平成 19 年度から令和元年にかけて約 15% も増加している。

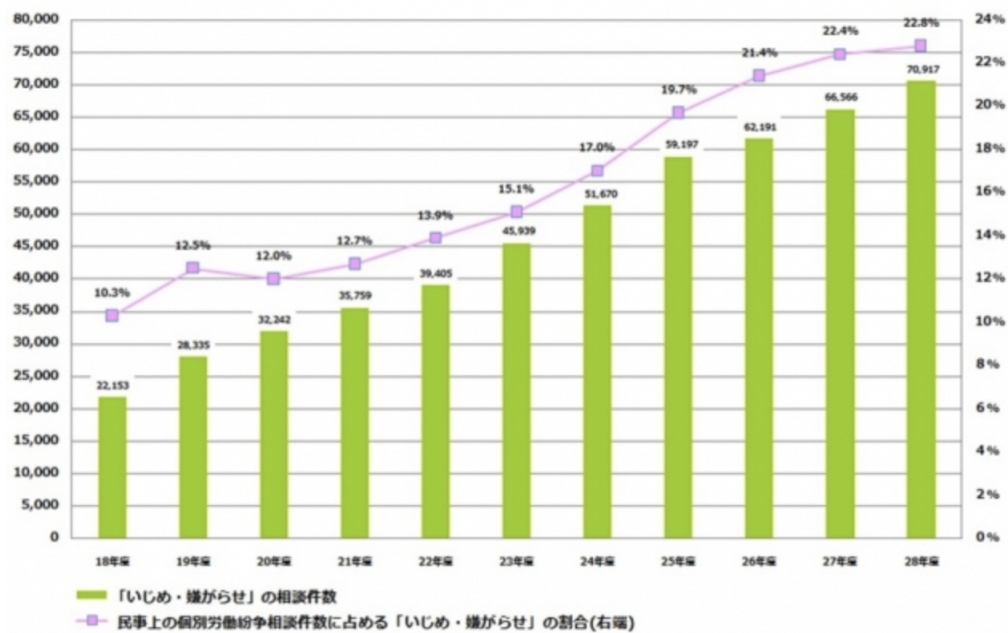


図 1.2 厚生労働省によるパワハラ調査

そこで我々は、以上の課題である「笑顔の減少」と「パワハラの増加」の解決を目的に FaceAPI を用いた表情分析を活用した労務管理支援システムを提案する。

1.2 論文の構成

本論文は以下のような構成になっている。

第 1 章では研究背景を述べる。

第 2 章では本研究で開発する労務管理支援システムの概要を述べる。

第 3 章では本研究で開発する労務管理支援システムの構成を述べる。

第 4 章では本研究で開発する労務管理支援システムの実装結果を述べる。

第 5 章では本研究のまとめを述べる。

第 2 章

機能設計

2.1 図の表示

第 3 章

システムの実装方法

3.1 システム構成

本システムは以下のような構成で実装を行う。システム構成図を図 3.1 に示す。
労務管理支援システムを React で Web アプリとして作成した。さらにデータベースに Cloud Firestore, ユーザー認証に Firebase Authentication, 表情分析に face-api.js を使用した。

次のセクションでは, それぞれのサービスについて説明する。

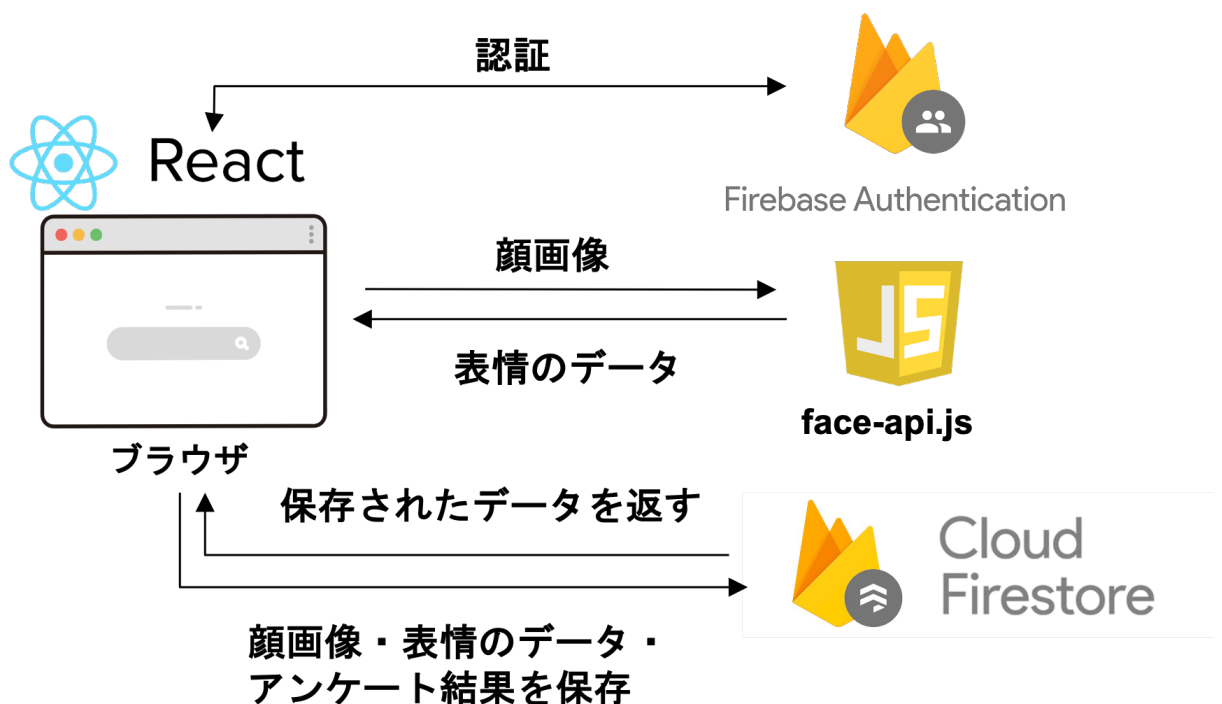


図 3.1 システム構成図

3.2 React について

React は、UI (ユーザインタフェース) 部分の構築に特化した JavaScript のライブラリで、React.js と呼ばれる。SNS で有名な Meta 社 (旧 Facebook 社) が自社サービスの機能拡張に伴うコードの複雑化によって維持管理がしにくくなることを防ぐために開発した。コーディングコストが少なく、開発規模が大きくなっても管理しやすいといった特長もあり、現在では開発元である Facebook 社のサービスである Facebook や Instagram はもちろんのこと、Yahoo! や Airbnb, Reddit, Netflix, Slack, Uber といった世界的な Web サイトや Web アプリで利用されるなど、世界中の多くの企業で採用されており、日本でも注目を集めるなど、今最も勢いのあるライブラリである。今回のシステム開発に React を採用した理由は三つある。

1. パフォーマンスが良い

React には、仮想 DOM (Virtual Document Object Model) というレンダリング機構が備わっている。仮想 DOM とは、実際の DOM ではなく、React 内部に持っている DOM の情報である。React を使うと、この仮想 DOM と実際の HTML 上の DOM を比較したときに出てくる違いだけが、毎回 HTML 上に再適用される。そのため画面全体が React で構成されていたとしても、必要な部分しか更新されず非常に高速に動作するため、パフォーマンスが良い。

2. UI コンポーネントのライブラリが多い

React は、世界中で使われているため、React のライブラリを使って UI をコンポーネント化するようになってきている。あらかじめ Button や Form などの UI パーツを React コンポーネントとして扱えるようにして、セット化したものが多くある。これらを使えば、今風の洗練された画面を作ることができる。

3. JavaScript の知識があれば使える

基本的に React は JavaScript で書かれているため、JavaScript の知識があればアプリを開発することができる。たとえ JavaScript の開発経験がなくても基本構文を理解していれば開発に取り掛かれる。今回、我々は JavaScript の学習を既に行なっていたので、React を選んだ。

3.3 face-api.js について

face-api.js はブラウザ, NodeJS で顔を検出するための, Tensorflow.js を活用した JavaScript API である. ここでは, face-api.js がどのような機能を提供しているかを紹介する.

- 顔検出
写真から顔を探して検出する.
- 顔のランドマーク検出
検出した顔の目や鼻の位置など, 顔の特徴を抽出する上で重要なキーポイントを検出する.
- 表情認識
検出した顔の表情を認識する. 表情は, 「怒り」, 「嬉しさ」, 「中立」, 「恐怖」, 「うんざり」, 「驚き」, 「悲しさ」の7種類あり, 数値で表される.
- 年齢推定
検出した顔の年齢を推定する.
- 性別認識
検出した顔の性別を推定する.

今回は, 顔検出と表情認識の機能を使い, 表情分析した値をグラフ化することを目的とする.

3.4 Material-UI について

Google 社が自社のサービスに統一的なデザインを与えるために作成した, デザインのガイドライン, または仕様書である Material Design を React のコンポーネントとして使えるようにしたものである. 本研究で Material Design を取り入れているのは, ユーザーにとってストレスが少なく, みやすく直感的に操作が出来るといった, 使いやすさを目的としているためである.

3.5 Firebase Authentication

Firebase Authentication はユーザー認証機能を提供し、ユーザ情報をクラウドで保存してくれる、Google 運営のサービスのことである。認証方法には、主に以下が用意されている。

- メールアドレスとパスワードによる認証
- 主要なプロバイダーアカウントによる認証
(Google / Twitter / Facebook / Github / Apple / Yahoo! / Microsoft 等)
- 匿名認証
- カスタム認証
- 電話番号認証

本アプリでは、メールアドレスとパスワードによる認証を使用してログインを行う。Firebase Authentication を使用した理由は、認証の安全性を外部に任せることができるからである。認証システムは、ユーザーの個人情報を預かることになるため、自前で実装するのは危険であり、万が一、個人情報が流出してしまうおそれがある。さらに、個人情報を保護して運用していくリスクとコストもかかってしまう。そこで外部ライブラリを使用することで、問題点を解決できると考えた。

本システムでは Firebase Authentication のソーシャルログイン機能として提供されている、Google サインインを利用してアプリケーションの認証システムを作成した。Google サインインはユーザーの Google アカウントを用いてアプリケーションのログインを可能にするシステムである。

ソーシャルログイン機能を利用することで、ユーザーは本アプリケーションの ID とパスワードを覚えておく必要がなくなる。また、ID とパスワードを自分で設定してログインするよりも手順が減るため、ユーザーにログインしてもらうハードルを低くすることができる。

3.6 Cloud Firestore

第4章

Tex の書き方 (応用編)

4.1 応用コマンド・環境

4.1.1 newcolumntype コマンド

表のカラムの書式を新規に割り当てることのできるコマンドです。表を作成する前に当コマンドで新しい書式を作成することで使用することが可能です。

「`\newcolumntype{書式名}{書式}`」のように記述することで書式を作成することができます。幅が 5cm で中央ぞろえのカラムを作成する場合には「`>{\centering\arraybackslash}p{5cm}`」を書式に記載します。

4.1.2 breakbox 環境

枠で囲まれた文章を作成することができます。図を囲む screen 環境と違い、ページをまたいだ枠を作ることができます。ソースコードを載せる場合等に使用するとよいでしょう。

|表示例|

breakbox 環境内の文章です。breakbox 環境内の文章です。breakbox 環境内の文章
です。breakbox 環境内の文章です。breakbox 環境内の文章です。breakbox 環境内
の文章です。breakbox 環境内の文章です。breakbox 環境内の文章です。breakbox
環境内の文章です。breakbox 環境内の文章です。breakbox 環境内の文章です。
breakbox 環境内の文章です。breakbox 環境内の文章です。breakbox 環境内の文章
です。

4.2 マクロ

Latex のマクロはコマンドと環境の両方を作成することが可能です。

4.2.1 コマンドマクロ

コマンドのマクロは「`\newcommand{コマンド名}{内容}`」のように記述することで作成することができます。引数を指定することもでき、「`\newcommand{コマンド名}[引数の数]{(#1) 内容}`」のように記述します。

|記述例|

```
\newcommand{\hoge}[2]{#1 は #2 である}
```

作成したコマンドを使用する場合は「`\hoge{我輩}{猫}`」のように記述します。

|表示例|

作成したコマンドを使用する場合は「我輩 は 猫 である」のように記述します。

4.2.2 環境マクロ

環境のマクロは「`\newenvironment{環境}[引数の数]{はじめ}{おわり}`」のように記述することで作成することができます。コマンドと同じく引数を指定できます。

|記述例|

```
\newenvironment{fig}{
    \begin{figure}[!h]
    \begin{screen}
    \begin{center}
}{
    \end{center}
    \end{screen}
    \end{figure}
}
\begin{fig}
    \includegraphics[scale=0.4, clip]{./img/apple.png}
    \caption{図の名称}
    \label{fig: 図の名称}
\end{fig}
```

|表示例|



図 4.1 図の名前

謝辞

|記述例|

本研究に関しまして，熱心かつ丁寧にご指導いただきました，千葉工業大学情報科学研究科情報科学専攻中村直人教授に心から御礼申し上げます。また，修士論文発表における，副査を務めていただきました浮貝雅裕教授，須田宇宙准教授に深謝いたします。そして，中村研究室博士前期課程，学部生の皆様，および，卒業された博士前期課程，学部生の皆様に心から御礼申し上げます。皆様のおかげで，これまでの研究生生活を充実かつ楽しく送ることが出来ました。そして，これまで支えてくれた両親をはじめとする親族各位に改めて御礼申し上げます。ありがとうございました。

参考文献

- [1] Web デザイン編集委員会, “Web デザイン-コンセプトメイキングから運用まで-改訂版”, CG-ARTS 協会 (2013).
- [2] 三井田惇郎/須田宇宙, “数値計算法 第 2 版”, 森北出版株式会社 (2013).