



夢ある未来を、共に創る。

# IoT、エッジコンピューティングで こんなことをやってみました(おかんの好きなIoT)

サーバ技術部 第一課  
(チーム名：ミルクボール)

2020年 1月 22日 (水)

1. 本資料の目的及び参画者
2. 顔認証とは
3. OpenCVを選定した理由
4. こんなことをやってみました
5. テスト環境の説明
6. テスト内容
7. まとめ

# 1. 本資料の目的及び参画者

### ■ 目的

「オカンがな、... オカンがな、好きなモノの特徴を忘れたらしくてね」

「どんな特徴やったか教えて」

「なんでも、食べ物 何とかパイ みたいな名前のやつを使って人を判別するやつらしいで」

「なあんや それやったらRaspberry Pi 4+カメラ+オープンソフトウェアを組み合わせたものでできるIoTというもののやがな」

**入手しやすいパーツのよせ集めで構築可能な顔認証システムでどこまでできるか？！**

※本資料は外部資料を流用している部分が多いため、社内限定の閲覧としてください。

### ■ 参加メンバー

重行 明 (サーバ技術部 第1課 / 北浜)

水野 安信 (サーバ技術部 第1課 / 北浜)

矢野 貴久 (サーバ技術部 第1課 / 中部)

木地 昌風 (サーバ技術部 第1課 / 平和島)

川杉 喜彦 (サーバ技術部 第1課 / 平和島)

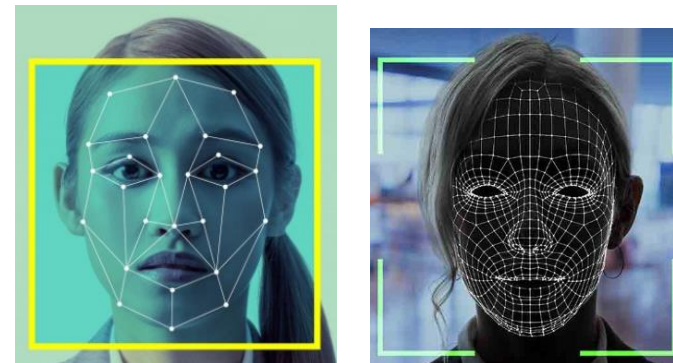
オブザーバ: 中村(匠)さん、中村(僚)さん

※恥ずかしいけどな、中村(匠)さん、中村(僚)さん 以外は全員IoT初心者やがな。

## 2. 顔認証とは

### 顔認証とは

我々が普段相手を判別する手段をITで実現した最も身近な認証方式である。顔の目、鼻、口などの特徴点の位置や顔領域の位置や大きさをもとに照合を行う。**なりすましが困難**なためセキュアであり、物理的なカギを持ったり、パスワードを設定する必要がない。利便性に優れるなどの特長を持っている。



### 利用分野

空港の税関やテーマパーク(USJ,富士急ハイランドなど多数)、コンサートなどで入場するための本人認証で使われている。



USJ 以下のNECサイトより抜粋  
<https://jpn.nec.com/info-square/mitatv/discover/02/2.html>

### 3. OPENCVを選定した理由

1. まずはRaspberry Pi + センサーの組み合わせで何かできないか意見招集を実施。



2. 技術発表会まであと2ヶ月しかないという事で、急ぎ Raspberry Pi と カメラ を調達しつつ、みなさんにわかりやすいものをお届けにするにはやはり画像処理系が良いということでカメラを使うもので顔認証のシステム辿り着いた。



3. Raspberry Pi と顔認証の組み合わせで情報量が多いOpenCVを選定した。

※[顔認証 Raspberry Pi]でググると先頭に OpenCVがヒットする。



## OpenCV(正式名称: Open Source Computer Vision Library)

OpenCVとは

オープンソースのライブラリです。

コンピュータで画像や動画処理するのに必要な、さまざまな機能が実装されており、BSDライセンスで配布されていることから学術用途だけでなく商用目的でも利用できる。加えて、マルチプラットフォーム対応されているため、幅広い場面で利用されている。

OpenCVを使うと、主に以下のような機能を利用できる。

- オブジェクト追跡(Object Tracking)
- 領域分割(Segmentation)
- カメラキャリブレーション(Calibration)
- 特徴点抽出
- 物体認識(Object recognition)
- 機械学習(Machine learning)
- パノラマ合成(Stitching)
- コンピュータショナルフォトグラフィ(Computational Photography)
- GUI(ウィンドウ表示、画像ファイル、動画ファイルの入出力、カメラキャプチャ)

OpenCVは画像処理から始まり顔認識、そして顔認証まで網羅している。

## 4. こんなことをやってみました どんなことがやれるか？

まずは案出しから行いました。

(案出し例)

1. 水田の水量と農業用水路の水量を同時に計測(水位センサー)、  
稲穂のたれ具合を見る(カメラ: 農家による目視)
2. 家の猫や子供の監視(カメラやセンサーを総合的に使って実現)  
猫の場合⇒家の中に赤外線カメラをつけて猫の生活場所が分かる  
⇒ヒートマップで確認できるとか時系列でどこにいるのか把握する  
子供: 冷蔵庫の開閉回数(ドア開閉センサー)、玄関の開閉回数(ドア開閉センサー)。  
子供: 子供の間食具合や外出状況などを把握する
3. 複数のカメラを使用して特定の人物を追跡。  
遊園地のアトラクション行列の先頭と最後尾にカメラを設置し、特定の人物が通過するのにかかった時間から待ち時間を算出。
4. などなど。。。

結果:

様々は案が出たものの、2ヶ月という期間で、結果を出す必要がある。。。   
ということで、カメラを用いた**顔認証システム**に落ち着いた。

## 5. テスト環境の説明

技術二課が構築したインフラを用いて、顔認証 について  
こんなことをやってみました

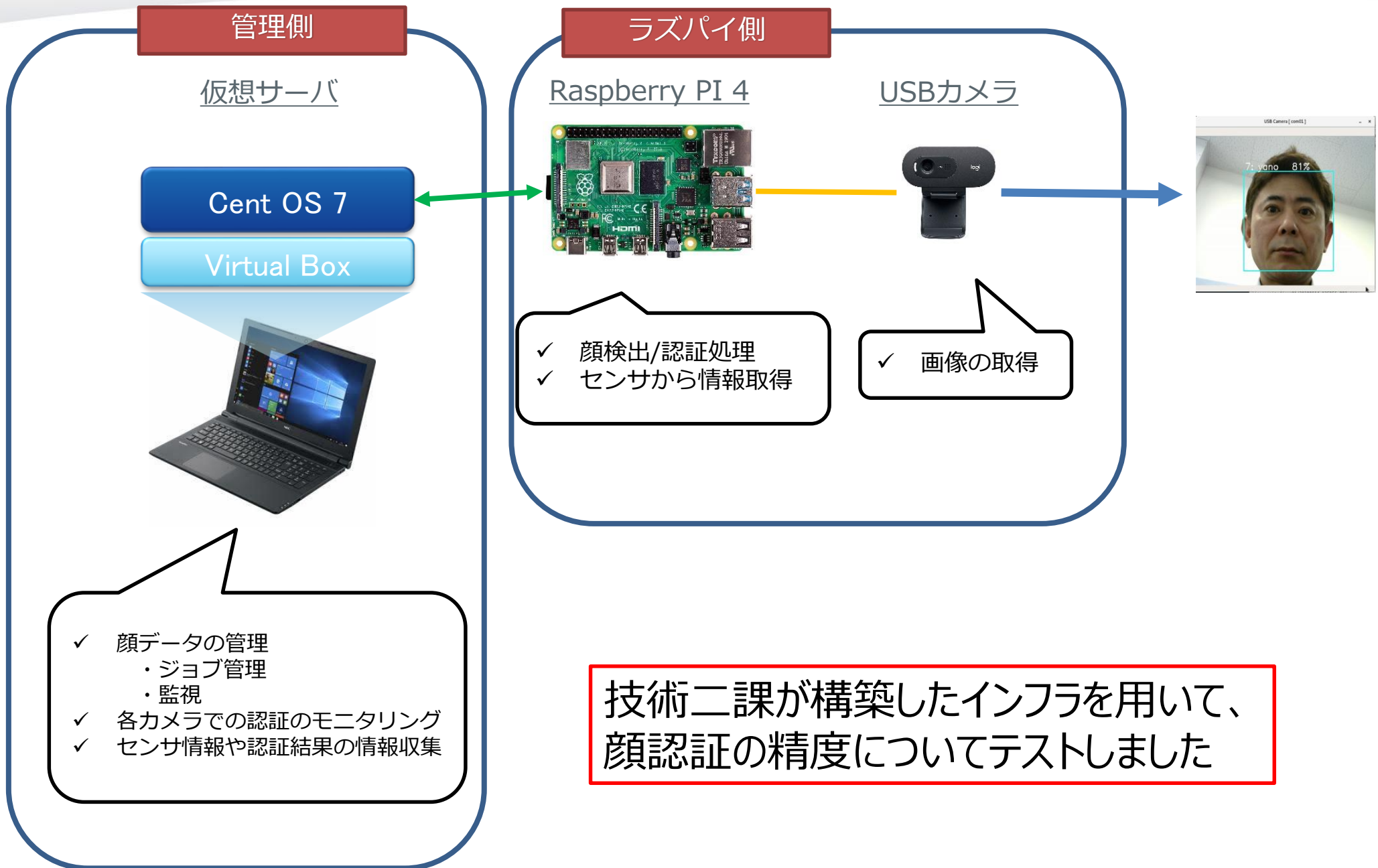
### ■ 技術二課が構築した環境について

#### おさらい

- Raspberry Pi 4 : ¥9000-  
ケース、SDカード含めて1万ちょっとで購入できます。4世代目になりメモリは4GBオンボード搭載している。  
もちろん、技適マークの認証もあります。
- カメラ : ¥2000-  
ラズパイ専用モジュールでなくUSBタイプを利用した。
- OS/CentOS7 : ¥0-  
Pi4までサポートしているARM対応OSは公式のRaspbian(32bit)、CentOS(32bit)、Ubuntu(32/64bit)
- ライブラリ/OpenCV : ¥0-  
インテルが開発・公開したオープンソースのコンピュータビジョン向けライブラリである

こんな金額ベースで実施しました！

※同ページは技術二課資料より抜粋

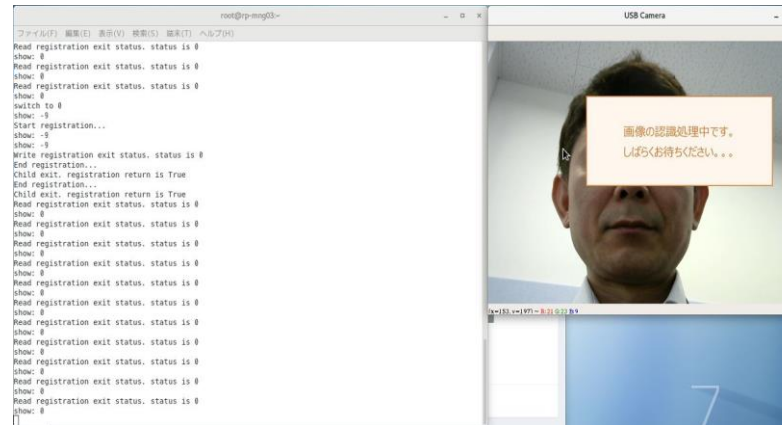


## 6. テスト内容

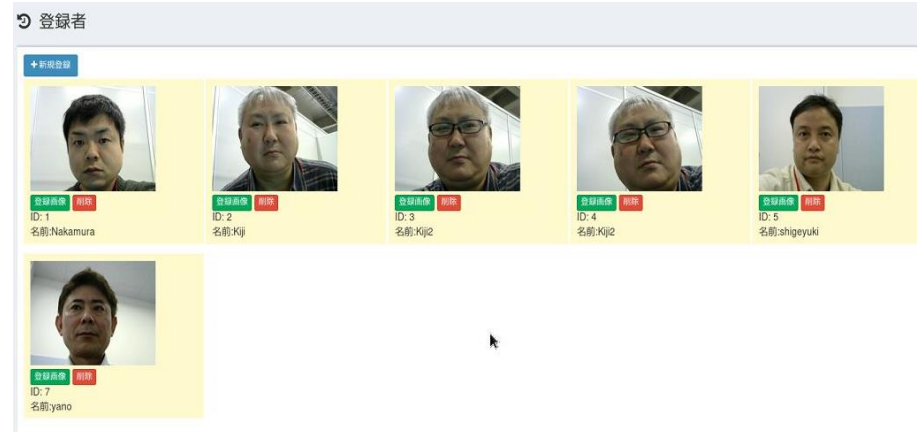
こんなことをやってみました

### 1. 顔登録 3人 それぞれにID を割り当てる

#### ① 顔画像を撮影



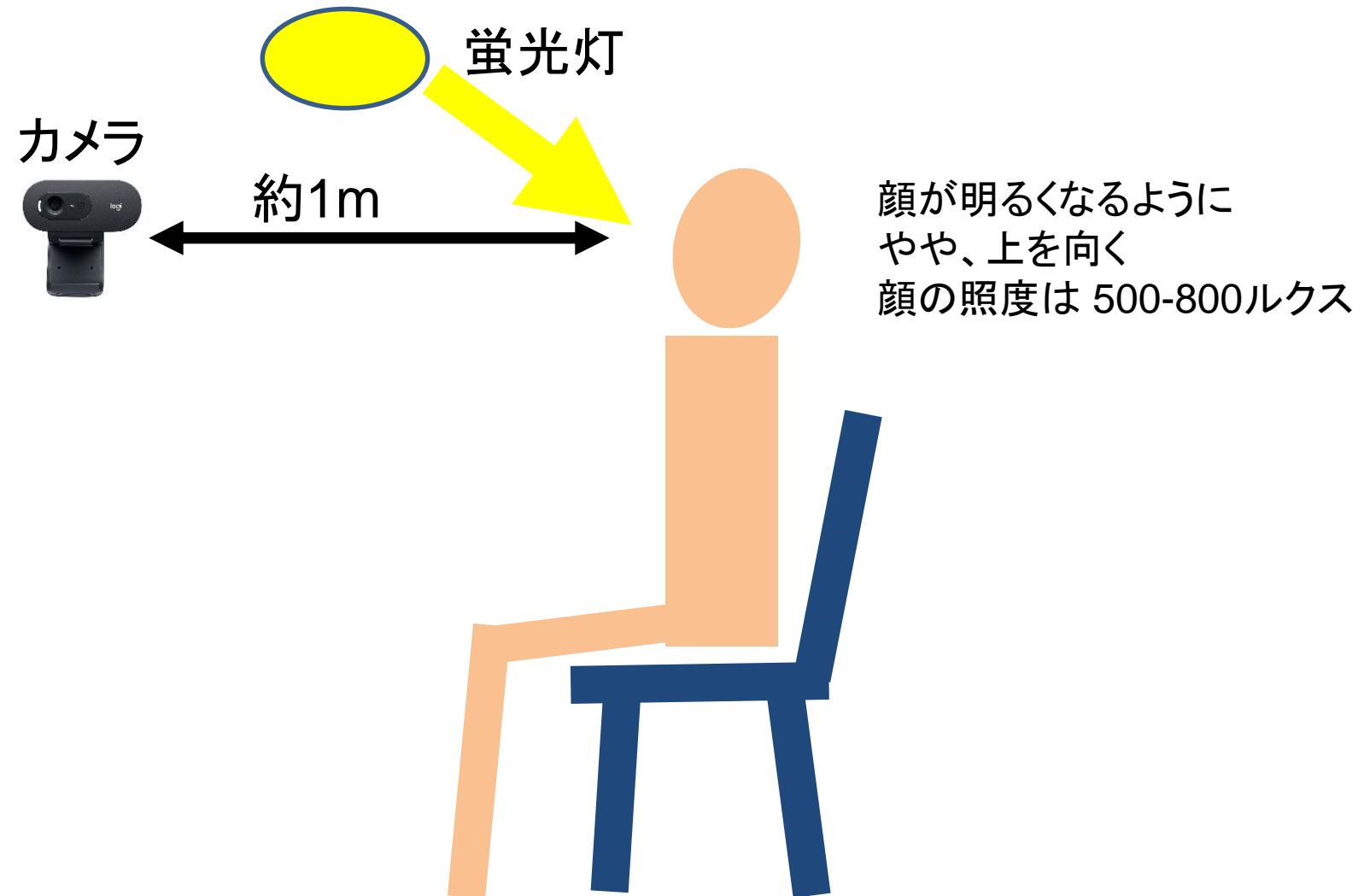
#### ② 顔画像データを登録





## 2. 顔認証

- ①IDを入力
  - ②ラズパイのカメラ正面に座り、撮影
  - ③640x480@10フレーム/秒 で撮影し、ディスクに画像ファイルとして保存
- ※ラズパイの処理能力により処理が停止するため上記とした。



### 3. 2の内容を以下のパターンで繰り返し、判定を確認する

パターンA：普段の表情

パターンB：口を開く(“い”を発声)

パターンC：目を閉じる

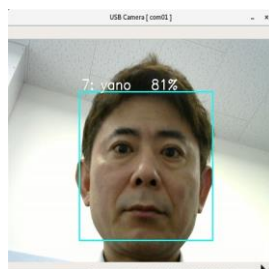
パターンD：別人

※ラズパイが即座にフレーム単位で判定する

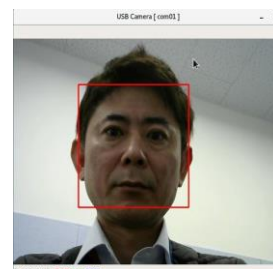
※1枚でも**70%以上**の合致であれば、認証OKとする  
認証率の妥当性が不明なため一旦上記とする

#### 顔認証

☐ 認証OK



☐ 認証NG



#### 【ログ】 ※ が認識OKのフレーム

認証ログ										
First	Previous	36	37	38	39	40	41	42	43	44
45	46	Next	Last							
検知日時 : 2020-01-14 14:02:40	検知日時 : 2020-01-14 14:02:40	検知日時 : 2020-01-14 14:02:40	検知日時 : 2020-01-14 14:02:40	検知日時 : 2020-01-14 14:02:40	検知日時 : 2020-01-14 14:02:40	検知日時 : 2020-01-14 14:02:40	検知日時 : 2020-01-14 14:02:40	検知日時 : 2020-01-14 14:02:40	検知日時 : 2020-01-14 14:02:40	検知日時 : 2020-01-14 14:02:40
ID : 9	ID : 9	ID : 9	ID : 9	ID : 9	ID : 9	ID : 9	ID : 9	ID : 9	ID : 9	ID : 9
名前: yano3	名前: yano3	名前: yano3	名前: yano3	名前: yano3	名前: yano3	名前: yano3	名前: yano3	名前: yano3	名前: yano3	名前: yano3
カメラ: com01	カメラ: com01	カメラ: com01	カメラ: com01	カメラ: com01	カメラ: com01	カメラ: com01	カメラ: com01	カメラ: com01	カメラ: com01	カメラ: com01
認証率: 77%	認証率: 77%	認証率: 78%	認証率: 78%	認証率: 78%	認証率: 78%	認証率: 78%	認証率: 78%	認証率: 78%	認証率: 78%	認証率: 78%
検知日時 : 2020-01-14 14:02:40	検知日時 : 2020-01-14 14:02:40	検知日時 : 2020-01-14 14:02:39	検知日時 : 2020-01-14 14:02:39	検知日時 : 2020-01-14 14:02:39	検知日時 : 2020-01-14 14:02:39	検知日時 : 2020-01-14 14:02:39	検知日時 : 2020-01-14 14:02:39	検知日時 : 2020-01-14 14:02:39	検知日時 : 2020-01-14 14:02:39	検知日時 : 2020-01-14 14:02:39
ID : 0	ID : 0	ID : 0	ID : 0	ID : 0	ID : 0	ID : 0	ID : 0	ID : 0	ID : 0	ID : 0
名前: yano3	名前: yano3	名前: yano3	名前: yano3	名前: yano3	名前: yano3	名前: yano3	名前: yano3	名前: yano3	名前: yano3	名前: yano3
カメラ: com01	カメラ: com01	カメラ: com01	カメラ: com01	カメラ: com01	カメラ: com01	カメラ: com01	カメラ: com01	カメラ: com01	カメラ: com01	カメラ: com01
認証率: 74%	認証率: 74%	認証率: 69%	認証率: 69%	認証率: 61%	認証率: 61%	認証率: 61%	認証率: 61%	認証率: 59%	認証率: 59%	認証率: 59%
検知日時 : 2020-01-14 14:02:39	検知日時 : 2020-01-14 14:02:39	検知日時 : 2020-01-14 14:02:39	検知日時 : 2020-01-14 14:02:39	検知日時 : 2020-01-14 14:02:39	検知日時 : 2020-01-14 14:02:39	検知日時 : 2020-01-14 14:02:39	検知日時 : 2020-01-14 14:02:39	検知日時 : 2020-01-14 14:02:39	検知日時 : 2020-01-14 14:02:39	検知日時 : 2020-01-14 14:02:39
ID : 0	ID : 0	ID : 0	ID : 0	ID : 0	ID : 0	ID : 0	ID : 0	ID : 0	ID : 0	ID : 0
名前: yano3	名前: yano3	名前: yano3	名前: yano3	名前: yano3	名前: yano3	名前: yano3	名前: yano3	名前: yano3	名前: yano3	名前: yano3
カメラ: com01	カメラ: com01	カメラ: com01	カメラ: com01	カメラ: com01	カメラ: com01	カメラ: com01	カメラ: com01	カメラ: com01	カメラ: com01	カメラ: com01
認証率: 61%	認証率: 62%	認証率: 60%	認証率: 60%	認証率: 59%	認証率: 59%	認証率: 60%	認証率: 60%	認証率: 60%	認証率: 60%	認証率: 60%

## 6. テスト内容 こんなことをやってみました

	パターン	認証度(%)
木地 ID8	普段の表情	78
	口を開く(“い”発音)	66
	目を閉じる	76
	別人: 矢野	51
	別人: 重行	51
矢野 ID9	普段の表情	76
	口を開く(“い”発音)	65
	目を閉じる	75
	別人: 重行	53
	別人: 木地	51
重行 ID10	普段の表情	77
	口を開く(“い”発音)	74
	目を閉じる	74
	別人: 矢野	63
	別人: 木地	52

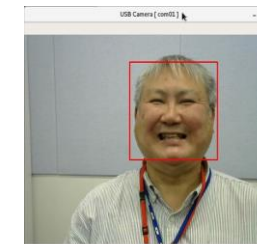
### 登録者



- ・木地(50代・男性)
- ・矢野(40代・男性)
- ・重行(40代・男性)



普段の表情



口を開く(い発音)



目を閉じる

## 7. まとめ

### 【考察と展望】

#### ■ 考察

- ・当初の想定通り、照度がある程度高いほど認証率は高い傾向にあった。
- ・ラズパイの処理能力の関係で低解像度 & 低フレームレート(640x480 & 10フレーム/秒)としたため精度は低いと思われる。今後ラズパイの性能向上により、この問題が改善される可能性がある。
- ・本人での表情変化と他人での認証度の組み合わせで同程度の認証度となってしまった事象や同じ画像での認証度でも80%程度となるためOpenCV自体の精度の問題も顕著になった。このあたりはOpenCV自体のソフトウェアの成熟度を待つ必要があるのかも知れない。そのため商用利用には現状程遠いことが理解できた。

### 【考察と展望】

#### ■今後のサービスの利用について

- ・ラズパイのような汎用的な装置の組み合わせで、我々のようなIoT初心者でも環境構築ができることがわかった。
- ・ラズパイには、温度・湿度・照度・音声など、さまざまなセンサーがあり、IoTをPoCするにはもってこいの装置である。組み合わせにより新たなサービスを創出することも可能だと考えている。
- ・例えば、次ページにつづく……

	案	詳細	実現性	IoT度
1	水田の水量と農業用水路の水量を同時に計測	(水位センサー)稲穂のたれ具合を見る(カメラ:農家による目視)	△	△
2	家の猫や子供の監視(カメラやセンサーを総合的に使って実現)	猫の場合⇒①水が減った量(水位センサー)、キャットフードの食べ具合(重量センサー)、 ⇒時間でデータを取得して、健康状態の管理に役立てる	△	△
3		猫の場合⇒②家の中に赤外線カメラをつけて猫の生活場所が分かる ⇒ヒートマップで確認できるとか時系列でどこにいるのか把握する	△	△
4		子供⇒冷蔵庫の開閉回数(ドア開閉センサー)、玄関の開閉回数(ドア開閉センサー)。 ⇒子供の間食具合や外出状況などを把握する	△	△
5	リアルタイムで落書き	カメラで撮影した映像を画像編集アプリ(snowとか)でリアルタイム落書き	△	△
6	店の入り口にカメラを置いて、顔認証で客層を解析	⇒年齢層や男女のデータを集めて、時間帯による客層について解析する	△	△

## 7. まとめ

	案	詳細	実現性	IoT度
7	新商品の前に配置して顔認証で客層を解析	⇒年齢層や男女のデータを集めて、時間帯による客層について解析する	△	△
8	バス車内にカメラを設置し、顔認証	乗客：体調が悪い状態を検出	△	△
9		運転手：体調の良し悪し、居眠り、スマホ遊びなどを検出	△	△
10	定点観測カメラを近隣、遠隔地それぞれに設置し、気象情報を比較する。	例) 沖縄は雨が多い	○	△
11	リモートワークの効率化/自動ログインの実現	ラズパイ1号機のWebカメラに顔を向け検知する ブラウザでSCSKのVPNサイトが開く ラズパイ2号機のWebカメラでブラウザを検知する 社員番号とVPNパスワードが自動入力される。	△	△
12	業務効率化：フリーデスク誘導システム	ラズパイ1号機のWebカメラに顔を向け検知する(通路を歩いている場面) フリーデスクの空いている席のLEDを点滅させる ラズパイ2号機のWebカメラに顔を向け検知する ⇒ラズパイ1号機で合致した画像の場合、ブラウザ認証 ⇒ラズパイ1号機で合致しない画像の場合、許可させてない席である警告orLED点滅orアラーム音	△	△



	案	詳細	実現性	IoT度
13	倉庫やデータセンターへの共連れ防止と作業監視	商品倉庫などで、登録されている作業員だけで作業行われている事を確認する。また、登録されていない他者がもぐりこんでいないかを確認する。	△	△
14	複数のカメラを使用して特定の人物を追跡。	遊園地のアトラクション行列の先頭と最後尾にカメラを設置し、特定の人物が通過するのにかかった時間から待ち時間を算出。アプリなどに反映することで客がリアルタイムで待ち時間の確認が可能。実現可能な場合どんな行列にも応用できそう	△	△
15	カメラで本人確認または年齢確認。	無人レジの会計時に顔認識技術で本人確認or年齢確認を実施。年齢確認の必要な商品(タバコや酒)の認証に使用	△	△
16	太陽光発電の候補地の具体的な日照時間を計測し可視化するSystem	照度計のセンサを配置したraspberry piを複数台用意して太陽光発電を行う候補地の発電容量をPoCする	○	○

「ほんまにわかれへんがな、そのパイどうなってんねん。」

「オトンがいうには、パイの塩焼きちゃうかって。」

「いや、絶対ちゃうやろ！」

「もうええわ。どうもありがとうございました。」

